

# A Learning Strategy for Multi-robot Based on Probabilistic Evolutionary Algorithm

Jiancong Fan<sup>1</sup>, Yongquan Liang<sup>1</sup>, Jiahong Ruan<sup>2</sup>

<sup>1)</sup> College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong, China  
(fanjiancong@sdust.edu.cn)

<sup>2)</sup> Scientific Research Department, Shandong Jiaotong University, Jinan 250023, China

**Abstract**—Estimation of distribution algorithm (EDA) is a new evolutionary computation method based on probabilistic theory. EDA can select optimal individuals through estimating probability distribution function of a population. The capture problem among multi software robots can be solved by EDA. The capture problem involves that some pursuers pursue several evaders through part of trajectory. The trajectory was produced by the evaders during their two-dimensional random mobility. The pursuers estimate the evaders' mobility functions and adjust their pursuit models to capture the evaders as fast as possible. The probabilistic evolutionary courses of multi-robot experiencing some competitions are analyzed in performances. The analysis shows that capture problem of multi-robot solved by EDA is better than other methods in several aspects.

**Keywords**—evolutionary computation, estimation of distribution algorithm, multi-robot competition, capture course

## 基于概率演化算法的多机器人竞争学习策略

樊建聪<sup>1</sup>, 梁永全<sup>1</sup>, 阮久宏<sup>2</sup>

<sup>1)</sup> 山东科技大学 信息科学与工程学院, 青岛, 山东, 中国

<sup>2)</sup> 山东交通学院 科研处, 济南, 山东, 中国

**摘 要** 分布估计算法作为一种新的演化计算方法, 通过估计概率分布函数以获取适应度高的个体及其分布, 并利用此分布函数产生新一代种群。利用分布估计算法求解多机器人之间的捕获问题, 多个追捕者捕获多个逃跑者。追捕者通过逃跑者已产生的部分移动轨迹及其在二维随机移动过程中的捕获参数, 估计逃跑者的移动函数。根据逃跑者的移动分布情况, 追捕者调整自身的追捕函数, 以尽快捕获逃跑者。通过多次竞争过程, 对多机器人的演化过程进行了性能上的分析, 表明基于分布估计算法捕获问题求解方法在时间和迭代次数上都优于其它方法。

**关键词** 演化计算, 分布估计算法, 多机器人竞争, 捕获过程

### 1. 一种新的演化方法——分布估计算法

分布估计算法(Estimation of Distribution Algorithm, 缩写为 EDA)是一类演化计算方法, 其主要特点是生成新一代种群时不需要设计交叉和变异算子, 而是通过估计一个概率分布函数来显式描述种群中适应度较高的个体及其分布情况, 并利用这个分布函数进行随机采样生成新一代种群。

也就是说, 与传统的演化算法相比, EDA 利用从被选择的个体估计的概率分布进行抽样产生新的个体, 取代了利用交叉和变异算子产生新个体的过程。EDA 的概念最初在 1996 年提出[9], 在 1999 年以后迅速发展, 成为当前演化计算领域前沿的研究内容, 近年来国际上演化计算领域的各大学术会议, 如 ACM SIGEVO、IEEE CEC 等, 都将分布估计算法作为重要专题予以讨论。

根据概率模型的复杂程度以及不同的采样方法, 分布估计算法发展了很多不同的具体实现方法, 但是都可以归

山东省自然科学基金项目(资助号: Y2007G07)和山东科技大学春蕾计划资助(资助号: 06540040512)

纳为下面两个主要步骤[10]:

(1) 构建描述解空间的概率模型。通过对种群的评估,选择优秀的个体集合,然后采用统计学习等手段构造一个描述当前解集的概率模型。

(2) 由概率模型随机采样产生新的种群。一般的,采用蒙特卡罗方法,对概率模型采样得到新的种群。

## 2. 捕获问题

捕获(PE, Pursuit-evasion)问题主要研究如何控制一群机器人(本文采用软件机器人,即 Agent)追捕一个或几个逃跑 Agent 及相关问题的领域,典型的例子很多,如搜索与营救行动,追捕敌军任务等,也可以推广到其它领域,如通过用户留下的上网痕迹和记录,获取用户兴趣特征、机器人追赶移动的目标,抓捕网络入侵者等。PE 包括确定的和不确定的两类问题。确定性的 PE 将追捕者和逃跑者的活动区域被抽象为一个有限连通图,节点表示 PE 参与者可以到达的位置,边表示行动路线,包括已经采取的行动和将要采取的行动。不确定的 PE 是有一个确定的具有节点的路线图,只是事先假定 PE 区域先验概率图,通过将连续空间 F 分解为多个有限数量的子区域  $R_i$ ,  $R_i$  作为节点,节点之间是否需要相连取决于不确定 PEG 的设计方法,例如概率方法通过概率计算来获取需要哪些连接以形成图 G,使得在 F 上的行动转化为在 G 中的搜索[3,4]。由 F 获得 G 的过程非常耗时,而且计算量较大,学习得到的图 G 对于 PEG 的 F 来说也不一定准确,为解决这一问题,Thrun 等人提出了通过最大似然估计法来获得一个图[5],以及后来出现的概率框架,将图的学习过程与追捕过程结合并同时进行的方法[6-8]。

关于 Agent 的 PE 问题的研究可以分为几类。一类是将 PE 问题抽象为一个图或树,追捕 Agent 通过图搜索的形式捕获逃跑 Agent。Barrière 等把搜索入侵者的过程抽象为加权树的结构,设计了一个线性时间复杂度的算法[1];第二类是在网络中追捕入侵者,这方面的文献较多[3-6]。这类方法将网络入侵者作为 evading agents 进行检测和追捕,不涉及搜索区域的问题;第三类是通过建立模型,如概率模型[1]、协议策略等。Chen 等人通过使用合同网交互协议而建立了 Multi-Agent 通信协议,并且将 agents 分为 pursuit agents 和 evading agents,利用协议使得 evading agents 向 pursuit agents 分布少的区域移动[2]。

本文利用分布估计演化算法,给出了一种新的软件机器人追捕-逃跑竞争学习策略。使用此策略可以用较少的迭代次数以较快的速度完成竞争过程。本文第 3 节介绍了相关概念和给定部分运动轨迹条件下的学习算法;第四节给

出了算法的运行过程和相关性能的比较与分析。

## 3. 给定部分运动轨迹条件下的算法

### 3.1 多软件机器人的移动方式

把追捕者  $P=\{p_1, p_2, \dots, p_n\}$  和逃跑者  $E=\{e_1, e_2, \dots, e_m\}$  的活动区域投影为一个二维坐标平面, P 和 E 以质点的形式在坐标平面上表示。P 和 E 中元素的位置以坐标的形式表示。为了求解方便,将坐标离散化为边长为  $\epsilon(\epsilon>0)$  的正方形区域,  $p_i(i=1,2,\dots,n)$  与  $e_j(j=1,2,\dots,m)$  在相邻接的区域间移动,把这种移动称为二维随机移动。

定义 1 在二维坐标平面 XOY 中, (X,0)和(0,Y)称为移动反射壁垒。

定义 2 在二维坐标平面 XOY 中, Agent 自主移动过程中, 如果符合以下条件:

- (1) Agent 一次只能移动到相邻接的单元格;  
在随机情况下,
- (2) 如果 Agent 的当前位置有一侧是反射壁垒, 则 Agent 有  $k(k=3 \text{ 或 } 5)$  个可选的移动方向, 以  $1/k$  的概率向邻接区域移动;
- (3) 如果 Agent 的当前位置的边界都不是反射壁垒, 则 Agent 以  $1/8$  的概率向邻接的各个方向移动;  
在 Agent 具有偏好的情况下,
- (4) Agent 以

$$\lambda_i P(d_i)$$

概率向邻接区域移动。其中  $\lambda_i$  称为偏好系数, 是 Agent 向某个或某些方向移动的习惯或认知倾向的度量。  $d_i$  表示移动的方向。在这种情况下, 当 Agent 邻接反射壁垒时

$$\sum_{i=1}^3 \lambda_i P(d_i) = 1 \text{ 或 } \sum_{i=1}^5 \lambda_i P(d_i) = 1$$

否则

$$\sum_{i=1}^8 \lambda_i P(d_i) = 1$$

则称这种移动为二维随机移动。

在本文中给出如下假设: 逃跑者所经过的位置具有标记 T,  $T=\langle \text{Token}, t \rangle$ , 其中 Token 是标记符,  $t$  表示时间, 称为时间戳。根据此假设, 定义逃跑方程:

$$\begin{aligned} f(X(T_{i+1}.t)) &= f(X(T_i.t) + 1) \\ f(X(T_n.t)) &= f(X(T_m.t)) + (n - m) \end{aligned}$$

$$\text{其中, } T_n.t > T_m.t$$

由于逃跑者的轨迹具有一定的随机性, 很难直接确定整个轨迹的曲线函数, 因此本文通过局部移动轨迹来获取局部移动函数, 并且按时间求解这些局部移动函数, 来获

取逃跑者的当前位置。

**定义 3 局部移动函数：**在 PE 过程中，从逃跑者所经历的所有轨迹中，以分段的方法连续截取部分轨迹，以更好的确定这些部分轨迹的曲线函数。这些部分轨迹曲线称为局部移动函数。

### 3.2 移动轨迹的度量方法

**定义 1 相近函数：**给定一个正整数  $\varepsilon$ ，在某个实数区间  $[-\theta, +\theta]$  上， $\theta \geq 0$ ，两个函数  $f_1$  和  $f_2$  的距离  $|f_1 - f_2| < \varepsilon$ ，则称  $f_1$  和  $f_2$  为相近函数。

**定义 2 相似函数：**设  $f_1$  是目标函数， $f_2$  是候选函数， $f_1$  与  $f_2$  在相同参数下其自变量的概率分布是相似的或者相同的，则称  $f_1$  与  $f_2$  为相似函数。

本文通过分布估计方法选择优化的运行轨迹，因此将相近函数和相似函数统称为近似函数。

相似函数和近似函数实际上都是描述目标模型和候选模型之间的相似程度。如果两个相近函数  $f_1$  与  $f_2$  的距离在任意区间上都为 0，或者相似的两个函数的概率分布完全一样，在这种极端情况下  $f_1$  与  $f_2$  是同一函数或模型。也就是说相近函数是从数值距离的角度，相似函数是从概率分布的角度来定义两个或多个函数之间的近似度量。

由于 Bhattacharyya (巴氏)系数是一种散度型测量标准，其几何意义是两个向量角度的 cosine 值，因此本文采用 Bhattacharyya 距离作为相似函数的度量。假设目标模型(函数)为  $q(x)$ ，候选模型(函数)是  $p(x)$ ，根据 3.1 节的二维随机移动过程， $q(x)$ 和  $p(x)$ 的 Bhattacharyya 距离定义为

$$D_B(p, q) = -\ln\left(\sum_{x \in X} \sqrt{p(x)q(x)}\right)$$

假设追捕者和逃跑者的移动范围符合多元高斯分布  $p_i = N(m_i, P_i)$ ，其中  $m_i$  和  $P_i$  分别是分布的均值和误差，则

$$D_B = \frac{1}{8}(m_1 - m_2)^T P^{-1}(m_1 - m_2) + \frac{1}{2} \ln\left(\frac{\det P}{\sqrt{\det P_1 \det P_2}}\right)$$

其中  $P = (P_1 + P_2) / 2$

令  $\rho = \sum_{x \in X} \sqrt{p(x)q(x)}$ ， $\rho$  越大，则 DB 越小，两个模型

$p(x)$ 和  $q(x)$ 越相似。

### 3.3 基于概率分布的竞争学习策略

在追捕逃跑过程中，追捕者的局部移动函数首先确定为目标函数  $q_0$ ，通过逃跑者的移动轨迹确定其局部移动函数作为候选函数  $p_0$ 。设时间步长为  $\tau$ ，即从时刻 0 开始，每间隔时间  $\tau$  就根据已存在的移动轨迹计算追捕者与逃跑者的 Bhattacharyya 距离。若距离小于某个阈值，则追捕者调

整追捕策略，向着 DB 值更小的分布函数调整。

根据 3.1 节给出的逃跑方程，将每个参与者的局部移动模型作为个体，从多个竞争 Agent 中取 1 个追捕者和 1 个逃跑者，令  $q$  是追捕者的移动函数， $f$  是逃跑者的移动函数，表 1 说明了一个简单的分布估计演化过程。

表 1 分布估计演化计算表

| $i$      | E        | P        | $D_B(q, f)$ |
|----------|----------|----------|-------------|
| 1        | $f_{e1}$ | $q_{p1}$ | $D_1$       |
| 2        | $f_{e2}$ | $q_{p2}$ | $D_2$       |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$    |
| $i$      | $f_{ei}$ | $q_{pi}$ | $D_i$       |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$    |
| $n$      | $f_{en}$ | $q_{pn}$ | $D_n$       |
| $p_0(x)$ |          |          |             |
| $p_1(x)$ |          |          |             |

在表 1 中， $p_j(x)$ 表示第  $j$  代中局部移动函数相似的概率。 $DB(q, f)$ 表示移动函数  $q$  和  $f$  的巴氏距离。

假设每一代种群规模为  $n$ ，取正整数  $\xi$ ，从一代种群中选取所有  $D_j \leq \xi$  的个体(设为  $m$ )，按以下方案进行处理：

(1) 如果  $m=1$ ，选符合条件的个体作为新的种群，并从此个体开始进行新的移动；

(2) 当  $m=2$ ，如果是连续的两次移动，则取这两次连续轨迹为新的种群继续移动；否则，任取一个符合条件的个体按照第(1)步进行处理；

(3) 当  $m=3$ ，如果是连续的三次移动，则取这三次连续轨迹为新的种群继续移动；否则，按照第(2)步进行处理；

(4) 当  $m=k$ ，如果是连续的  $k$  次移动，则取这  $k$  次连续轨迹为新的种群继续移动；否则，按照第( $k-1$ )步进行处理；

在上述过程中有几点需要说明，一是种群中最优个体分布概率的计算，利用以下公式进行计算

$$p_t(x) = \frac{\text{count}(D_j \leq \xi)}{\text{count}(D_j)}$$

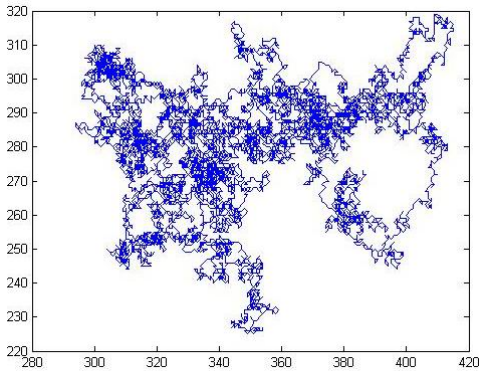
其中， $j \in \{1, 2, \dots, n\}$ ， $t$  表示第  $t$  代种群。二是  $\xi$  具有动态变化性，当  $P_t(x) < P_{t+1}(x)$ ，即第  $t+1$  代相似函数出现的概率大于第  $t$  代时， $\xi$  值下降，下降的幅度根据  $DB(q, f)$  的值确定。

## 4. 性能比较与分析

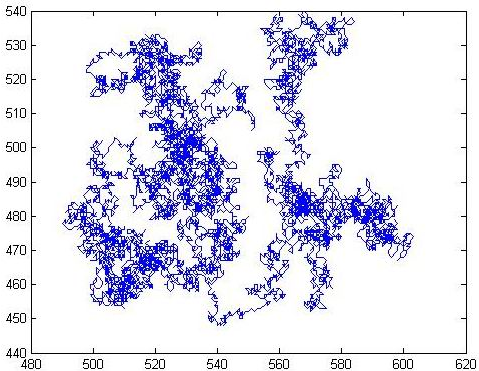
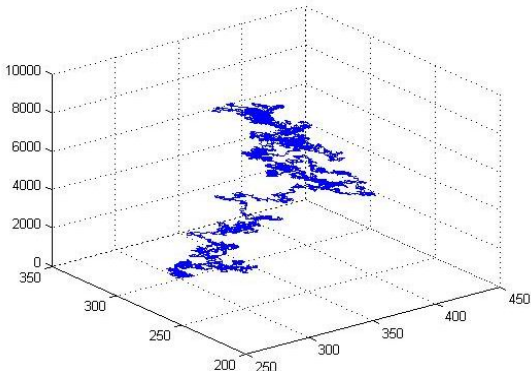
取一次移动轨迹的局部移动函数(由 104 个点组成的轨迹所产生的函数)，如图 1 所示。图 1(1)所示的是追捕 Agent 的二维和三维移动轨迹。图 1(2)和(3)所示的是两个逃跑 Agent( $\text{Agent}_1$  和  $\text{Agent}_2$ )的二维和三维移动轨迹。可以通过 Matlab 模拟出这些移动轨迹的近似函数，但由于巴氏距离中的目标函数和候选函数可以是轨迹点，所以本文不需要拟合出轨迹点的函数表达式，而是直接使用它们的轨迹点

文件计算巴氏距离。这样可以更加真实的反映出局部移动函数之间的距离。实际上图 1 所示的是追捕机器人和两个逃跑机器人的移动轨迹，取连续 10000 个移动估计点作为

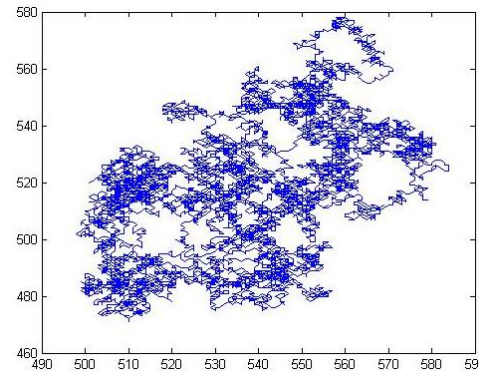
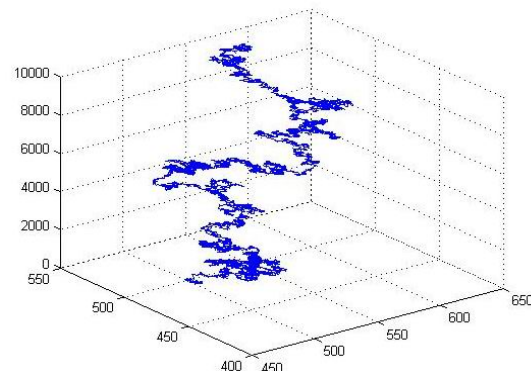
一个个体，可以计算出追捕者与逃跑者 1(图 1(2)所示)的 DB 值为 18.876，追捕者与逃跑者 2(图 1(3)所示)的 DB 值为 19.7195。



(1) 追捕者的一次移动轨迹(取连续的 10000 个点)



(2) 逃跑者 1 的一次移动轨迹(取连续的 10000 个点)



(3) 逃跑者 2 的一次移动轨迹(取连续的 10000 个点)

图 1 一次移动轨迹 ( $DB_{1,2}=18.8760$ ,  $DB_{1,3}=19.7195$ )

追捕者和一个逃跑者之间的连续 4 次移动轨迹作为一代种群，计算他们的 Bhattacharyya 距离，如表 2(a)所示。在表 2(a)中，令  $\xi=20$ ，则选择  $i=4$  和 5 作为最优个体继续移动，此时分布概率为 40%，第 2 代种群如表 2(b)所示，取  $i=6,7,8$  为新的个体，且分布概率为 60%， $\xi$  变为 19，第 3 代种群如表 2(c)所示，取  $i=13,14,15$  为新的个体，且分布

概率为 60%， $\xi$  取 18，第 4 代令  $\xi$  为 10，则  $i=17,18,19,20$  作为新的种群个体，此时分布概率为 80%。

图 2 所示的是经过 50 代演化的轨迹分布，追捕者和逃跑者的分布相似度 DB 为 2.7390，它们的分布基本相同，追捕者可以根据轨迹分布情况捕获逃跑者。

表 2 经过 4 代的一次演化过程

(a) 第一代( $\zeta=20$ )

| $i$ | E        | P        | $D_B(q_s, f)$ |
|-----|----------|----------|---------------|
| 1   | $f_{e1}$ | $q_{p1}$ | 18.8760       |
| 2   | $f_{e2}$ | $q_{p2}$ | 19.7632       |
| 3   | $f_{e3}$ | $q_{p3}$ | 43.5261       |
| 4   | $f_{e4}$ | $q_{p4}$ | 15.2320       |
| 5   | $f_{e5}$ | $q_{p5}$ | 18.8562       |

(b) 第二代( $\zeta=19$ )

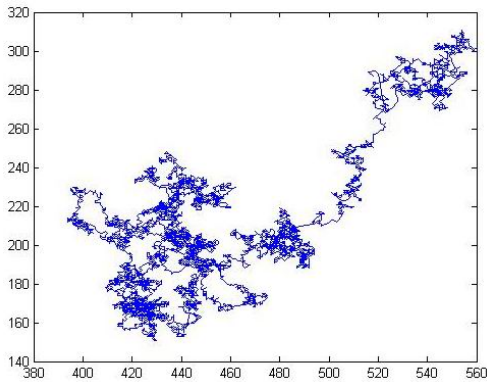
| $i$ | E         | P         | $D_B(q_s, f)$ |
|-----|-----------|-----------|---------------|
| 6   | $f_{e6}$  | $q_{p6}$  | 15.2320       |
| 7   | $f_{e7}$  | $q_{p7}$  | 18.8562       |
| 8   | $f_{e8}$  | $q_{p8}$  | 10.0433       |
| 9   | $f_{e9}$  | $q_{p9}$  | 26.4652       |
| 10  | $f_{e10}$ | $q_{p10}$ | 19.0065       |

(c) 第三代( $\zeta=18$ )

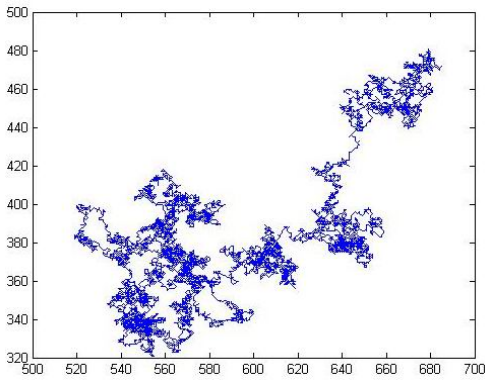
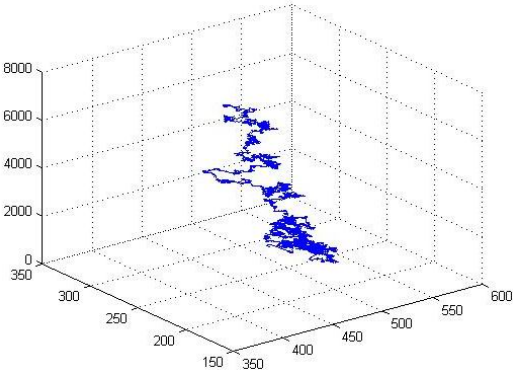
| $i$ | E         | P         | $D_B(q_s, f)$ |
|-----|-----------|-----------|---------------|
| 11  | $f_{e11}$ | $q_{p11}$ | 15.2320       |
| 12  | $f_{e12}$ | $q_{p12}$ | 18.8562       |
| 13  | $f_{e13}$ | $q_{p13}$ | 10.0433       |
| 14  | $f_{e14}$ | $q_{p14}$ | 8.4273        |
| 15  | $f_{e15}$ | $q_{p15}$ | 7.3832        |

(d) 第四代( $\zeta=10$ )

| $i$ | E         | P         | $D_B(q_s, f)$ |
|-----|-----------|-----------|---------------|
| 16  | $f_{e16}$ | $q_{p16}$ | 10.0433       |
| 17  | $f_{e17}$ | $q_{p17}$ | 8.4273        |
| 18  | $f_{e18}$ | $q_{p18}$ | 7.3832        |
| 19  | $f_{e19}$ | $q_{p19}$ | 5.4357        |
| 20  | $f_{e20}$ | $q_{p20}$ | 7.0219        |



(a) 追捕者的一次移动轨迹(取连续的 10000 个点)



(b) 逃跑者 1 的一次移动轨迹(取连续的 10000 个点)

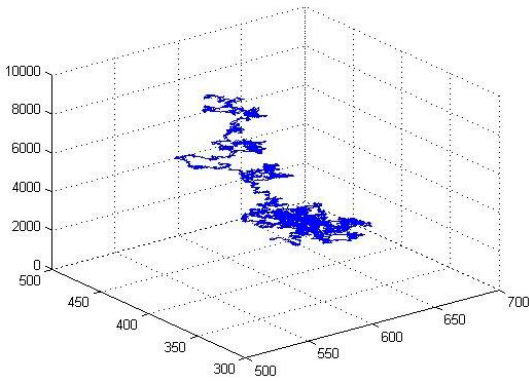


图 2 经过  $n$  次迭代后的移动轨迹 ( $D_{B1,2}=2.7390$ )

采用本文提出的策略与其它方法进行比较, 包括随机捕获和基于传统演化策略的捕获算法[11]。本文主要进行了

在不同种群个体选择(即取不同的连续移动次数)时, 三种策略的成功捕获次数的比较, 如表 4 所示。表 3 中是经过 50

次演化的平均值。

表 3 成功捕获次数比较表(50 次演化的平均值)

| 二维随机<br>移动区域 | random mobile |      | evolutionary algorithm |      | EDA ( $\zeta=1.0$ ) |      |
|--------------|---------------|------|------------------------|------|---------------------|------|
|              | succeed       | fail | succeed                | fail | succeed             | fail |
| 5000         | 1             | 49   | 5                      | 45   | 22                  | 28   |
| 10000        | 7             | 43   | 13                     | 37   | 36                  | 14   |
| 20000        | 17            | 33   | 29                     | 21   | 33                  | 17   |
| 50000        | 32            | 18   | 39                     | 11   | 45                  | 5    |
| 100000       | 37            | 13   | 45                     | 5    | 50                  | 0    |
| 200000       | 41            | 9    | 48                     | 2    | 50                  | 0    |

5. 总结

本文提出了基于分布估计算法的竞争学习策略。该策略利用分布概率获取最有可能发生的方案。将这种策略应用于多机器人追捕-逃跑的竞争问题上取得了较好的结果。分布估计算法是演化计算研究的新方向，目前其擅长的领域主要集中在函数的优化上，在求解函数极值方面表现出较好的性能[12-13]。概率是人工智能研究的基础工具之一，当然包括机器学习和机器人规划等领域，传统的演化计算以设计算子为主，没有充分考虑到概率在不确定问题的求解和推理过程中的重要性。因此分布估计算法必定会在学习算法和策略领域有广阔的应用空间。本文提出的竞争学习策略仅是利用分布估计算法求解问题的一个方面的反映，还有许多工作需要进一步研究和完善，如本文的实验分析仅是用软件 Agent 进行模拟，在一些方面进行了假设。在实际应用过程中可能有一些实践问题需要解决，这些都是我们下一步的工作。

参考文献

[1] T.D.Parsons. Pursuit-evasion in a graph. Theory and application of graphs, Berlin: Springer-Verlag, 1976: 426-441.

[2] N.Megiddo, S.L.Hakimi, M.R.Garey, D.S.Johnson, C.H.Papadimitriou. The complexity of searching a graph. Journal of the ACM,1988, 35(1): 18-44.

[3] A.S.Lapaugh. Recontamination does not help to search a graph. Journal of the ACM, 1993, 41(3): 224-245.

[4] I.Suzuki, M.Yamashita. Searching for a mobile intruder in a polygonal region. SIAM J.comput., 1992, 21: 863-888.

[5] S.Thrun, W.Burgard, D.Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. Machine Learning and Autonomous Robots, 1998, 31(5): 1-25.

[6] V.Isler, D.Sun, S.Sastry. Roadmap based pursuit-evasion and collision avoidance. Robotics: Science and Systems, 2005: 257-264.

[7] J.P.Hespanha, H.J.Kim, S.Sastry. Multiple-agent probabilistic pursuit-evasion games. Proceedings of the 38th IEEE Conference on Decision and Control, 1999: 2432-2437.

[8] R.Vidal, O.Shakernia, H.J.Kim, etc. Probabilistic pursuit-evasion games: theory, implementation and experimental evaluation. IEEE Transactions on Robotics and Automation, 2002,18(5): 662 -669.

[9] H.Mühlenbein, G.Paaß. From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. Lecture Notes In Computer Science; Vol. 1141, Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, London, UK, 1996, pp.178-187.

[10] M.Pelikan, D.E.Goldberg, F.Lobo. A Survey of Optimization by Building and Using Probabilistic Models. IlliGAL Report No. 99018, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, Illinois, 1999.

[11] Jiancong Fan, Jiuhong Ruan, Yongquan Liang. An Evolutionary Solution for Cooperative and Competitive Mobile Agents. Lecture Notes in Artificial Intelligence, Vol 5855, Proceedings of 2009 International Conference of Artificial Intelligence and Computational Intelligence, Shanghai, China, 2009, pp.599-607.

[12] 周树德, 孙增圻. 分布估计算法综述. 自动化学报, 2007, 33(2): 113-124.

[13] 董维山. 用于全局优化的分布估计算法. 博士学位论文, 2008.