

# 控制与决策

Control and Decision

## 基于改进人工鱼群算法和MAKLINK图的机器人路径规划

郭伟, 秦国选, 王磊, 孙日杰

引用本文:

郭伟, 秦国选, 王磊, 等. 基于改进人工鱼群算法和MAKLINK图的机器人路径规划[J]. *控制与决策*, 2020, 35(9): 2145–2152.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0030>

---

### 您可能感兴趣的其他文章

Articles you may be interested in

#### 基于多行为的移动机器人路径规划

Mobile robot path planning based on multi-behaviours

控制与决策. 2019, 34(12): 2721–2726 <https://doi.org/10.13195/j.kzyjc.2018.0278>

#### 基于文化萤火虫算法的足球机器人动态路径规划

Cultural firefly algorithm for dynamic path planning of soccer robot

控制与决策. 2018, 33(11): 2015–2020 <https://doi.org/10.13195/j.kzyjc.2017.0821>

#### 基于改进势场蚁群算法的机器人路径规划

Robot path planning based on improved ant colony algorithm with potential field heuristic

控制与决策. 2018, 33(10): 1775–1781 <https://doi.org/10.13195/j.kzyjc.2017.0639>

#### 基于A\*改进算法的四向移动机器人路径搜索研究

Research of four-way mobile robot path search based on improved A\*algorithm

控制与决策. 2017, 32(6): 1007–1012 <https://doi.org/10.13195/j.kzyjc.2016.0244>

#### 一种动态搜索策略的蚁群算法及其在机器人路径规划中的应用

Ant colony algorithm based on dynamic search strategy and its application on path planning of robot

控制与决策. 2017, 32(3): 552–556 <https://doi.org/10.13195/j.kzyjc.2016.0005>

# 基于改进人工鱼群算法和MAKLINK图的机器人路径规划

郭伟<sup>†</sup>, 秦国选, 王磊, 孙日杰

(天津大学机械工程学院, 天津 300354)

**摘要:** 针对静态二维环境下移动机器人全局路径规划问题, 提出一种基于改进人工鱼群算法 (IAFSA) 和 MAKLINK 图的路径规划方法. 该方法以 Lorentzian 函数和正态分布函数为视野和步长的自适应算子, 引入指数递减惯性权重因子, 能够提高 AFSA 算法的收敛速度和计算精度. MS (José Luis Esteves Dos Santos) 算法结合 IAFSA 算法分步寻优, 取 IAFSA 算法优化后的最优路径为全局最优路径, 可以解决以往算法在 MAKLINK 图中只能求近似全局最优路径的问题. 仿真实验结果表明了所提出改进算法方案的可行性和有效性.

**关键词:** 移动机器人; 路径规划; 人工鱼群算法; MAKLINK 图; MS 算法

中图分类号: TP242

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0030

**引用格式:** 郭伟, 秦国选, 王磊, 等. 基于改进人工鱼群算法和 MAKLINK 图的机器人路径规划 [J]. 控制与决策, 2020, 35(9): 2145-2152.

## Robot path planning based on improved artificial fish swarm algorithm and MAKLINK graph

GUO Wei<sup>†</sup>, QIN Guo-xuan, WANG Lei, SUN Ri-jie

(College of Mechanical Engineering, Tianjin University, Tianjin 300354, China)

**Abstract:** A global path planning method, based on the improved artificial fish swarm algorithm (IAFSA) and MAKLINK graph, is proposed to solve the global path planning problem in the two-dimensional static environment. Lorentzian function and normal distribution function are chosen as adaptive operators of step and visual, the exponential decreasing inertia weighting factor is also introduced, which can improve the convergence speed and accuracy of the AFSA algorithm. The MS algorithm is combined with the IAFSA algorithm to calculate for two steps. The optimal path optimized by the IAFSA algorithm is selected as the global optimal path, which solves the problem of that the previous algorithm can only get the approximate global optimal path in the MAKLINK graph. The simulation results show the feasibility and effectiveness of the proposed improved algorithm.

**Keywords:** mobile robot; path planning; artificial fish swarm algorithm; MAKLINK graph; MS algorithm

## 0 引言

路径规划是移动机器人导航技术的关键组成部分,其目的是依据某个或某些优化准则(如工作代价最小、时间最短、距离最短、能量最少等),在工作空间中找到一条从初始状态到目标状态的能避开障碍物的最优或接近最优路径<sup>[1]</sup>.目前,国内外学者对移动机器人路径规划已经做了大量的研究工作,提出或改进了很多算法,包括 A\* 算法<sup>[2]</sup>、Dijkstra 算法<sup>[3]</sup>、快速扩展随机树算法 (RRT)<sup>[4]</sup>、随机路标图算法 (PRM)<sup>[5]</sup>、人工势场法<sup>[6]</sup>等传统算法,以及遗传算法 (GA)<sup>[7]</sup>、蚁群算法 (ACO)<sup>[8]</sup>、粒子群算法<sup>[9]</sup>、神经网络算法<sup>[10]</sup>、模糊逻辑算法 (fuzzy logic)<sup>[11]</sup>、人工鱼群算法<sup>[12]</sup>等智能优

化算法.特别是包括蚁群算法、人工鱼群算法等在内的群体智能算法在移动机器人路径规划方面取得了很大的进步,获得了广泛关注.

人工鱼群算法 (artificial fish swarm algorithm, AFSA) 是一种新型群智能优化算法,模拟了自然界中鱼群的觅食、聚群、追尾、随机游动行为. AFSA 算法具有鲁棒性强、实现简单、寻优速度快、全局寻优能力强等优点,同时也存在搜索与开发平衡能力差、搜索后期收敛速度慢、不能求取精确值的缺点.文献 [13] 将对立强化学习概念引入到 AFSA 算法中,优化了人工鱼群在问题空间的分布,并在移动机器人 Pioneer 3-DX 进行了测试,实验表明改进的 AFSA 算

收稿日期: 2019-01-06; 修回日期: 2019-04-30.

基金项目: 国家社科重点基金项目 (16AZD004).

责任编辑: 方华京.

<sup>†</sup>通讯作者. E-mail: wguo@tju.edu.cn.

法路径规划能力强于A\*算法. 文献[14]使用基于对数函数的自适应视野、步长,提高了最优解的精度. 文献[15]将改进人工势场算法与AFSA算法相结合,仿真实验表明改进算法的最优路径搜索效率明显优于传统算法.

针对标准AFSA算法搜索后期收敛速度慢、不能求取精确值的问题,本文提出一种改进的人工鱼群算法(improve artificial fish swarm algorithm, IAFSA),算法采用自适应视野、步长,并引入指数递减惯性权重因子,大大提高了算法的收敛速度和计算精度. 针对以往算法在MAKLINK图中,只能求取近似全局最优路径的问题,将MS算法<sup>[16]</sup>与所提出的IAFSA算法相结合,MS算法首先计算前 $K$ 条最短路径,然后IAFSA算法快速优化这些路径,最后选取最短路径为移动机器人全局最优路径. 在MAKLINK图中,该方法的全局寻优能力明显强于传统方法.

## 1 基本算法描述

### 1.1 基本AFSA

AFSA算法模拟了鱼类的行为特征,自然界中鱼类通过觅食、聚集、追尾和随机游动来寻觅食物与同伴、躲避敌害,该过程对应算法中的4个行为:觅食行为、聚群行为、追尾行为和随机行为.

设在一个 $D$ 维搜索空间中有 $N$ 条人工鱼, $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ 表示第 $i$ 条人工鱼的状态, $i = 1, 2, \dots, N$ ;  $(x_{i1}, x_{i2}, \dots, x_{in})$ 表示第 $i$ 条人工鱼的优化变量; $Y = f(X)$ 表示人工鱼当前位置食物浓度(目标函数值);visual表示人工鱼的感知范围;step表示人工鱼移动最大步长; $|X_j - X_i|$ 表示第 $i, j$ 条人工鱼之间的欧氏距离; $\delta$ 表示拥挤度因子;try number表示人工鱼的最大试探次数.

#### 1.1.1 觅食行为

觅食行为是鱼类最基本的一种行为,处于 $X_i$ 位置的鱼在视野范围内随机确定一个位置 $X_j$ , $X_j$ 确定如下:

$$X_j = X_i + \text{visual} \times \text{rand}(0, 1), \quad (1)$$

其中 $Y_i, Y_j$ 分别为 $X_i, X_j$ 位置的食物浓度,若 $Y_j > Y_i$ ,则 $AF_i$ 向位置 $Y_j$ 方向移动一步,有

$$X_i(t+1) = X_i(t) + \frac{X_j(t) - X_i(t)}{\|X_j(t) - X_i(t)\|} \times \text{step} \times \text{rand}(0, 1), \quad (2)$$

$t$ 为迭代次数.

#### 1.1.2 聚群行为

聚群是鱼类的一种常见现象,为了趋利避害,大量的鱼聚集在一起,这是它们进化而来的一种生存方

式. 鱼群的中心为

$$X_c = \frac{1}{N} \sum_{i=1}^N X_i, \quad (3)$$

其中 $X_c$ 为鱼群状态的算术平均数. 定义 $n_f$ 为 $AF_i$ 视野内鱼的数目,若 $n_f/N < \delta$ 且 $Y_c > Y_i$ ,则认为 $X_c$ 位置食物浓度大且不拥挤, $AF_i$ 朝鱼群中心移动,即

$$X_i(t+1) = X_i(t) + \frac{X_c(t) - X_i(t)}{\|X_c(t) - X_i(t)\|} \times \text{step} \times \text{rand}(0, 1). \quad (4)$$

#### 1.1.3 追尾行为

一个或几个鱼发现食物时,其他邻近的伙伴会快速地跟随到达食物地点. $AF_i$ 的状态为 $X_i$ , $AF_i$ 视野中体格最大伙伴的状态为 $X_j$ , $n_f$ 为 $AF_i$ 视野内鱼的数目,若 $Y_j > Y_i n_f \delta$ ,表明伙伴 $X_j$ 位置有较高的食物浓度并且不太拥挤,则 $AF_i$ 朝伙伴 $X_j$ 移动,有

$$X_i(t+1) = X_i(t) + \frac{X_j(t) - X_i(t)}{\|X_j(t) - X_i(t)\|} \times \text{step} \times \text{rand}(0, 1). \quad (5)$$

如果视野内没有伙伴或者伙伴的状态都不满足要求,则 $AF_i$ 执行觅食行为.

#### 1.1.4 随机行为

随机行为也是鱼类最基本的一种行为,可以大范围地搜寻伙伴或食物,有效防止局部最优. 随机行为实现较为简单,就是在视野中随机选择一个状态,然后向该方向移动,只是觅食行为的一个缺省行为,有

$$X_i(t+1) = X_i(t) + \text{visual} \times \text{rand}(0, 1). \quad (6)$$

### 1.2 MS算法

MS算法由Martins等于2000年提出<sup>[16]</sup>,旨在高效率地计算有向图中前 $K$ 条最短路径. 定义有向图 $G(N, A)$ , $N$ 为节点集合, $A$ 为连接节点的弧集合, $A = (i, j)$ , $i, j$ 分别为连接弧 $A$ 的首末节点,则有向图 $G(N, A)$ 中连接节点 $s, t$ 的路径可以表示为

$$p = \{s, (s, i), i, \dots, j, (j, t), t\}, \quad (7)$$

连接节点 $s, t$ 的前 $K$ 条最短路径可以表示为

$$P = \{p_1, p_2, \dots, p_K\}, \quad (8)$$

其中 $d(p_i) \leq d(p_{i+1}), i = 1, 2, \dots, K-1, d(p_i)$ 为路径 $p_i$ 的总长度,且 $p_i$ 先于 $p_{i+1}$ 确定.

MS算法首先在有向图 $G_1(N_1, A_1)$ 中利用Dijkstra算法计算节点 $s, t$ 间的最短路径 $p_1$ ,“删除”路径 $p_1$ (这里的删除不是真的删除,而是通过有向图中增加附加节点和相应的弧来实现)得到 $G_2(N_2, A_2)$ ;然后Dijkstra算法计算 $s, t$ 在 $G_2(N_2, A_2)$ 中的最短路径 $p_2$ ,这里得到的 $p_2$ 为 $G_1(N_1, A_1)$ 中第2最短路

径, 在  $G_2(N_2, A_2)$  “删除” 路径  $p_2$  得到  $G_3(N_3, A_3)$ , ..., 以此类推, 直至计算到路径  $p_K$ ; 最后得到的  $\{p_1, p_2, \dots, p_K\}$  即为节点  $s$ 、 $t$  在有向图  $G_1(N_1, A_1)$  中的前  $K$  条最短路径。

## 2 改进人工鱼群算法

在基本人工鱼群算法中, 视野和步长固定不变, 视野决定了搜索范围, 步长决定了迭代速度和精度。搜索前期, 较大的视野和步长使鱼群快速接近全局最优值附近, 但接近目标时, 较大的视野和步长使鱼群错过食物浓度较高的位置, 这种现象越接近目标越明显, 表现为基本人工鱼群算法难以求得高精度的数值解。反之, 若使用较小的视野和步长, 则会降低算法的迭代速度和全局搜索能力, 增大陷入局部最优的可能性。为使人工鱼群算法前期保持较高的迭代速度, 后期保持较高的解精度, 本文提出了一系列改进措施。

### 2.1 自适应视野

为了平衡算法的全局搜索和局部搜索能力, 引入一个 3 参数 Lorentzian 函数<sup>[17]</sup> 改进视野为自适应视野, 即

$$f(x; x_0, \gamma, I) = I \left[ \frac{\gamma^2}{(x - x_0)^2 + \gamma^2} \right], \quad (9)$$

其中  $I$  为波峰值。视野进行自适应更新, 有

$$\text{visual}_{t+1} = \text{visual}_t \times f\left(\frac{4 \times t}{T_{\max}}; 0, 2, 1\right). \quad (10)$$

其中:  $t$  为目前迭代的次数,  $T_{\max}$  为最大迭代次数,  $\text{visual}_t$  为当前视野值,  $\text{visual}_{t+1}$  为下一次迭代视野值。在算法初始阶段, 视野值比较大, 利于全局最优值的搜索和加快迭代速度; 在算法后期, 视野值比较小, 利于局部搜索和提高解的精度。

### 2.2 自适应步长

为了平衡迭代速度和解精度, 引入如下变形的正态分布函数改进步长:

$$g(x) = \exp(-\pi x^2), \quad (11)$$

步长为

$$\text{step}_{t+1} = \text{step}_t \times g\left(\frac{t}{T_{\max}}\right). \quad (12)$$

其中:  $\text{step}_t$  为当前步长值,  $\text{step}_{t+1}$  为下一次迭代的步长值。算法迭代初期, 步长缓慢减小, 利于加快迭代速度; 算法迭代后期, 步长迅速减小, 利于提高解精度。

### 2.3 惯性权重因子

在基本人工鱼群算法中,  $\text{AF}_s$  通过执行 4 个基本行为来移动位置, 如式(2)~(6)所示。  $\text{AF}_i$  下一次迭代的位置由两部分组成: 上一次迭代的位置  $X_i(t)$  和随机步长  $\text{step} \times \text{rand}(0, 1)$ 。该策略会引起一个问题: 算

法迭代初期,  $X_i(t)$  导致算法全局搜索能力下降; 算法迭代后期,  $\text{step} \times \text{rand}(0, 1)$  导致算法局部搜索能力下降。针对该问题, 结合粒子群算法 (PSO) 的权重策略, 本文提出一种指数递减惯性权重因子对基本人工鱼群算法进行改进, 即

$$\beta = \beta_{\text{end}} \left( \frac{\beta_{\text{start}}}{\beta_{\text{end}}} \right)^{1 - \frac{t}{T_{\max}}}. \quad (13)$$

其中:  $t$  为目前迭代的次数,  $T_{\max}$  为最大迭代次数,  $\beta_{\text{start}}$  为第 1 次迭代时的权重,  $\beta_{\text{end}}$  为最后一次迭代时的权重,  $\beta$  为一个递减函数, 通过  $\text{step} \times \text{rand}(0, 1) \times \beta$ 、 $X_i(t) \times (1 - \beta)$  分别给两部分加上权重, 随着  $\beta$  逐步减小, 算法的全局搜索能力下降, 局部搜索能力上升。

### 2.4 新人工鱼群算法

对式(2)~(6)进行改进, 分别对应如下(其中:  $\text{step}_t$  为第  $t$  次迭代时的步长,  $\text{visual}_t$  为第  $t$  次迭代时的视野, 其他参数与基本人工鱼群算法相同):

1) 觅食行为

$$\begin{aligned} X_i(t+1) = & \\ (1 - \beta) \times X_i(t) + \beta \times & \frac{X_j(t) - X_i(t)}{\|X_j(t) - X_i(t)\|} \times \\ \text{step}_t \times \text{rand}(0, 1). & \end{aligned} \quad (14)$$

2) 聚群行为

$$\begin{aligned} X_i(t+1) = & \\ (1 - \beta) \times X_i(t) + & \\ \beta \times \frac{X_c(t) - X_i(t)}{\|X_c(t) - X_i(t)\|} \times & \text{step}_t \times \text{rand}(0, 1). \end{aligned} \quad (15)$$

3) 追尾行为

$$\begin{aligned} X_i(t+1) = & \\ (1 - \beta) \times X_i(t) + & \\ \beta \times \frac{X_j(t) - X_i(t)}{\|X_j(t) - X_i(t)\|} \times & \text{step}_t \times \text{rand}(0, 1). \end{aligned} \quad (16)$$

4) 随机行为

$$X_i(t+1) = X_i(t) + \text{visual}_t \times \text{rand}(0, 1). \quad (17)$$

### 2.5 改进算法具体步骤

图 1 为改进算法的流程, 具体步骤描述如下。

**step 1:** 初始化参数。设置起始位置, 终止位置, 前最短路径数目  $K$ , 人工鱼数目  $N$ , 步长  $\text{step}$ , 视野  $\text{visual}$ , 最大试探次数  $\text{try number}$ , 拥挤度因子  $\delta$ , 最大迭代次数  $T_{\max}$ , 初始权重  $\beta_{\text{start}}$ , 终止权重  $\beta_{\text{end}}$  等。

**step 2:** 采用 MS 算法计算出前  $K$  条最短路径, 并从小到大排列成列表。

**step 3:** 从列表中选择第 1 条路径  $p_1$  进行二次优化, 具体操作如 step 4。

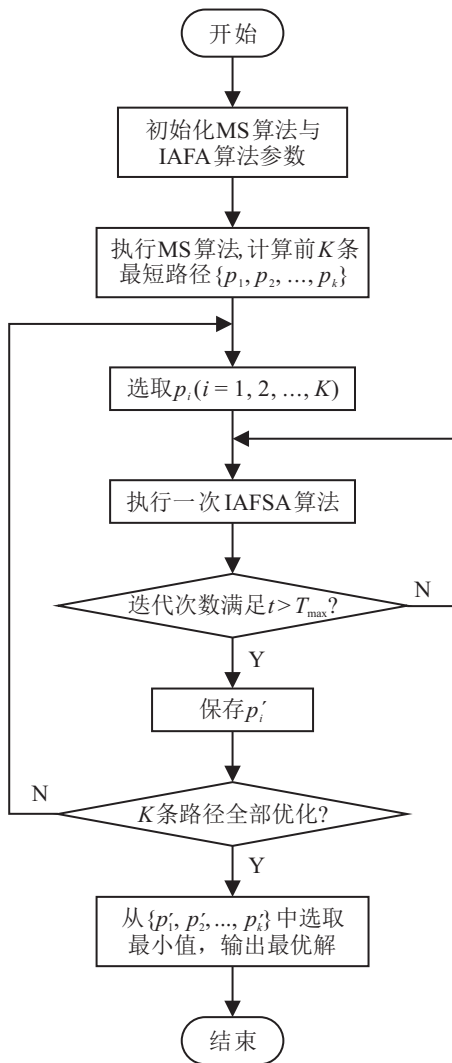


图1 改进算法的流程

step 4: 在人工鱼群中找到食物浓度值  $Y_{\max}$  对应的  $X_i$ , 并更新至布告板上,  $AF_s$  执行觅食、聚群、追尾、随机行为, 视野更新如式(10), 步长更新如式(12), 鱼群行为更新如式(14)~(17). 将每一次迭代得到的食物浓度值  $Y_{\max}$  与布告板上的食物浓度值进行比较, 若大于布告板上的食物浓度值, 则更新布告板.

step 5: 判断是否达到最大迭代次数或满足其他终止条件, 只要有一项条件满足, 则转至 step 6, 否则转至 step 4, 进行下一次迭代.

step 6: 返回最优解, 保存优化后路径  $p'_i$ .

step 7: 判断  $K$  条最短路径是否全部优化, 若满足条件, 则转至 step 8, 否则选取列表中下一条路径转至 step 4.

step 8: 从  $\{p'_1, p'_2, \dots, p'_K\}$  中选择最短路径, 输出最优值, 算法结束.

### 3 实验仿真与分析

为了验证改进算法的有效性和可行性, 进行大量仿真实验, 并通过与 MS-标准 AFSA、MS-文献[13]

IAFSA、A\*[2]、RRT[4]、PRM[5]、人工势场[6]、GA[7]、D-ACO[18]、fuzzy logic[11] 路径规划算法的对比, 验证本文算法的优越性. 算法运行环境为: Windows 7 64 bit, Matlab R2014a, 处理器 Intel(R) Core(TM) i5-3230 M, 主频 2.6 GHz, 内存 4 GB.

#### 3.1 地图建模

机器人路径规划的前提是环境建模, 通过相机、雷达等传感器提取到的信息, 分析计算得到机器人认识的环境地图, 使机器人在该环境中进行路径规划. 目前常用的环境建模方法有可视图法 (visibility graph)、链接图法 (MAKLINK graph)、栅格图法 (grids) 等. 栅格图法虽然应用广泛, 但存在复杂环境下环境信息存储量大、抗干扰能力弱、决策效率低下的缺点. 链接图法具有占用内存小、搜索复杂性低的优点, 因此本文采用链接图法建立地图.

链接图的建立基于以下假设[19]:

1) 多边形的高度平行于  $Z$  轴, 整个路径存在于  $XY$  平面;

2) 将障碍物的边界依据机器人的最大尺寸和机器人正常感知所需的最小范围进行扩展, 将机器人简化为一个质点.

障碍物用顶点表示, 假设第  $i$  个障碍物  $O_i$  有  $n_i$  个顶点, 整个环境可表示为

$$\begin{cases} O_i = \{(x_{1i}, y_{1i}), (x_{2i}, y_{2i}), \dots, (x_{ni}, y_{ni})\}, \\ W = \{WSB, O_1, O_2, \dots, O_m\}. \end{cases} \quad (18)$$

其中:  $O_i$  为第  $i$  个障碍物, WSB 为环境边界, 环境的改变只是  $W$  中元素的增减.

机器人环境中的自由空间是由自由链接线围成的凸区域构建的, 自由链接线满足 4 个条件:

1) 链接线的两端必须是两个多边形障碍物的两个顶点, 或者其中一个是障碍物顶点而另一个在环境的边界上, 在此意义下同一障碍物顶点的连线也计算在内;

2) 每条自由链接线是相邻两个自由凸区域的公共界线;

3) 自由链接线不能穿越环境中的任何障碍物;

4) 每个自由凸区域至少要有两条自由链接线作为边界.

将自由链接线的中点作为路径点, 路径点顺序为  $1, 2, \dots, n$ , 路径点的连线为机器人可自由移动的网络. 如图 2 所示, 设路径点  $S, T$  为起点和终点, 黑色多边形为障碍物, 细实线为自由链接线, 虚线为路径点连接而成的可供机器人自由移动的网络.

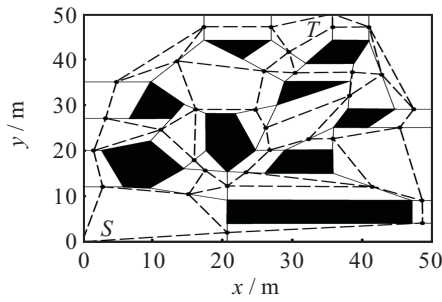


图2 MAKLINK图

3.2 前K条最短路径

按照正常思路,对于已经建立好的MAKLINK图,首先使用Dijkstra算法计算出“次优路径”,然后使用优化算法进行二次优化,得到“最优路径”,但是使用这种方法计算出的“最优路径”并不一定是真的最优路径<sup>[20]</sup>. 为此,本文提出首先使用MS算法计算前K条最短路径,然后使用本文提出的IAFSA算法对这K条路径进行二次优化,选取最短路径作为全局最优路径的策略,该操作大大提高了算法全局寻优的能力.

图2 MAKLINK图对应的35 × 35阶邻接矩阵为

$$\begin{bmatrix}
 \infty & 12.37 & \infty & \infty & \infty & \dots & \infty \\
 12.37 & \infty & 8.14 & \infty & \infty & \dots & \infty \\
 \infty & 8.14 & \infty & 7.28 & \infty & \dots & \infty \\
 \infty & \infty & 7.28 & \infty & 8.14 & \dots & \infty \\
 \infty & \infty & \infty & 8.14 & \infty & \dots & \infty \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
 \infty & \infty & \infty & \infty & \infty & \dots & \infty
 \end{bmatrix}. \quad (19)$$

图3为当K = 3,起点终点分别为S、T时,MS算法计算出的前3条最短路径.按照路径长短排序为路径1 < 路径2 < 路径3,长度分别为66.965 m、67.078 m、68.235 m.

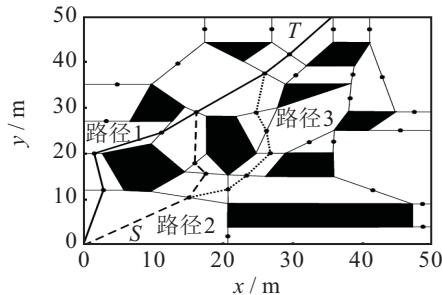


图3 MS算法计算前3条最短路径

3.3 IAFSA算法实现

假设由MS算法计算得到的最短路径为 $V_0, V_1, \dots, V_D, V_{D+1}$ ,其中 $V_0, V_{D+1}$ 分别为起点和终点,二次优化的工作便是调整 $V_1, \dots, V_D$ 的位置,使总体目标函数值最小,进而得到最优路径.具体的操作方式是 $V_i$ 在其自由链接线上滑动,优化后的路径点为

$$V_i = V_{i1} + (V_{i2} - V_{i1})t_i. \quad (20)$$

其中: $V_{i1}, V_{i2}$ 为 $V_i$ 所在自由链接线的两个端点, $t_i \in [0, 1], i = 1, 2, \dots, D$ (起点终点位置不滑动),每一组 $\{t_1, t_2, \dots, t_D\}$ 对应一条路径.

为了获得最短路径,将路径总长度作为IAFSA算法的目标函数,对应为

$$f(V) = \sum_{i=0}^D V_i V_{i+1}. \quad (21)$$

其中: $V_i V_{i+1}$ 为节点 $V_i, V_{i+1}$ 之间的距离, $V_0$ 为起点, $V_{D+1}$ 为终点.

3.4 仿真结果

IAFSA和AFSA算法参数设置如下: $N = 50, \delta = 0.618, \beta_{start} = 0.9, \beta_{end} = 0.2, \text{try number} = 10$ ,初始visual和初始step分别为搜索空间长度的30%与10%, $T_{max} = 100$ .

实验结果如图4所示:MS-本文IAFSA、MS-标准AFSA、MS-文献[13]IAFSA优化结果分别为

62.859 01 m, 64.826 96 m, 64.637 72 m;

3条路径对应的 $\{t_1, t_2, \dots, t_D\}$ 分别为

$\{0.073\ 634, 0.029\ 052, 0.281\ 085, 1.000\ 00,$

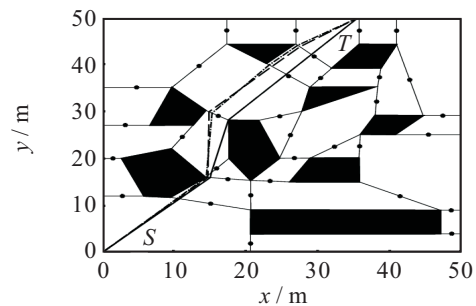
$0.513\ 572, 0.480\ 834\}$ ,

$\{0.046\ 471, 0.018\ 111, 0.047\ 003, 0.124\ 335,$

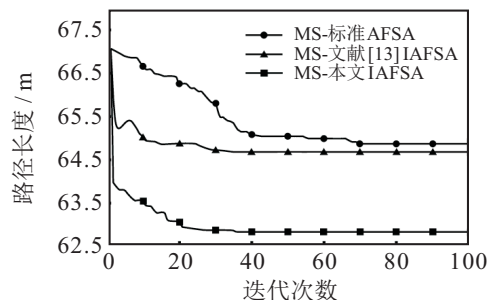
$0.143\ 754, 0.153\ 841\}$ ,

$\{0.044\ 974, 0.000\ 00, 0.066\ 571, 0.136\ 144,$

$0.057\ 864, 0.080\ 342\}$ .



(a) 路径规划结果对比



(b) 算法收敛对比

图4 3种人工鱼群算法仿真

可以看出,图4最优路径对应图3中的路径2,并不是路径1;二次优化效果明显,最优路径长度为62.859 01 m,明显小于MS算法算出最优路径长度66.965 m;MS-本文IAFSA算法收敛速度和计算精度远强于MS-标准AFSA算法,MS-本文IAFSA算法计算精度明显强于MS-文献[13]IAFSA算法,原因是文献[13]IAFSA算法采用的是线性递减惯性权重因子,本文IAFSA算法采用指数递减权重因子,在迭代后期,本文算法更容易靠近全局最优解。

表1为MS-本文IAFSA、MS-标准AFSA、MS-文献[13]IAFSA分别进行100次实验计算出的平均最优路径长度、达到收敛稳定时的平均迭代次数和平均运行时间.图5为起点终点分别为 $S_1、T_1、S_2、T_2、S_3、T_3$ 时由MS-本文IAFSA算法计算出的实验结果。

表1 3种人工鱼群算法100次重复实验结果对比

算法	最优路径长度/m	迭代次数	运行时间/s
MS-本文IAFSA	62.894 42	37	1.571 09
MS-标准AFSA	65.241 63	68	1.833 45
MS-文献[13]IAFSA	64.841 48	36	2.057 44

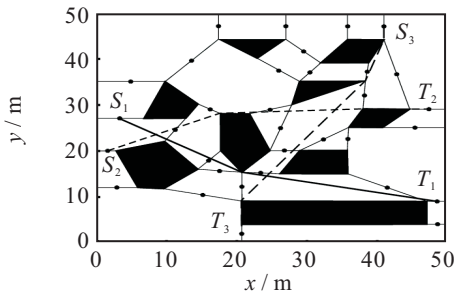
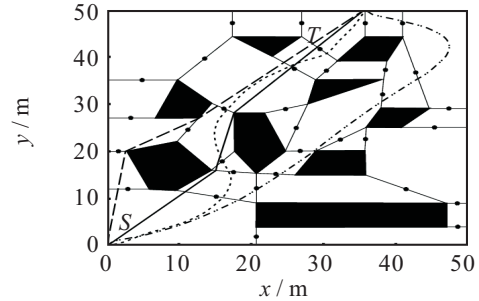


图5 对不同起始点优化后的路径

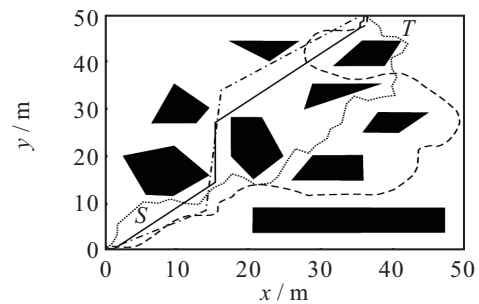
为进一步验证MS-本文IAFSA的全局优化能力,选取基于蚁群算法的二维路径规划算法(D-ACO<sup>[18]</sup>)、RRT<sup>[4]</sup>、PRM<sup>[5]</sup>、fuzzy logic<sup>[11]</sup>、A\*<sup>[12]</sup>、人工势场<sup>[6]</sup>、GA<sup>[7]</sup>算法与本文改进算法进行仿真实验比较.蚁群算法参数设置:迭代次数 $N_{C_{max}} = 100$ ,种群数量 $M = 50$ ,信息启发式因子 $\alpha = 1$ ,期望启发式因子 $\beta = 2$ ,全局信息素挥发系数 $\rho = 0.1$ ,信息素强度系数 $Q = 1$ .其他算法各参数设置与文献保持一致。

实验结果如图6所示.MS-本文IAFSA、D-ACO、GA、fuzzy logic优化路径分别对应图6(a)MAKLINK图中实线、虚线、点划线、点线,A\*、RRT、PRM、人工势场算法优化路径分别对应图6(b)普通地图中实线、点线、点划线、虚线.结合图6(c)和图6(d)可知,MS-本文IAFSA算法规划的路径长度明显小于其他算法,尤其是GA算法、人工势场算法,D-ACO算法

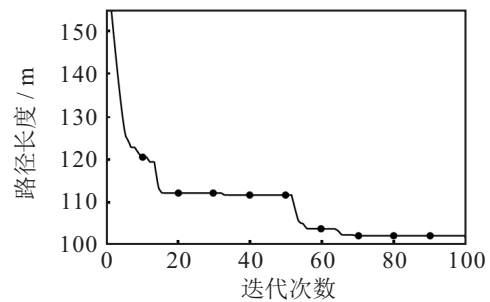
未找到最优路径,原因是D-ACO算法只在Dijkstra算法搜索到的最优路径上进行二次优化,没有考虑到其他次优路径.而MS-本文IAFSA算法是对MS算法搜索到的前 $K$ 条最短路径全部进行二次优化,取最优者为全局最优路径。



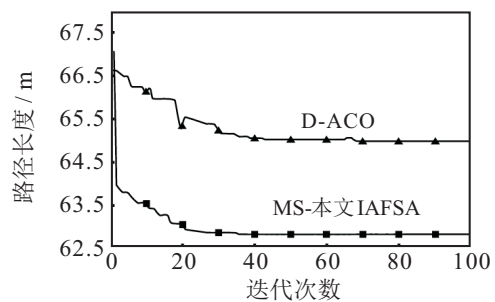
(a) MS-本文IAFSA、D-ACO、GA、fuzzy logic



(b) A\*、RRT、PRM、人工势场法



(c) GA收敛曲线



(d) MS-本文IAFSA、D-ACO收敛曲线

图6 各算法仿真比较

表2为各算法分别进行100次实验计算出的平均最优路径长度、达到收敛稳定时的平均迭代次数和平均运行时间.可以看出,相比于D-ACO算法之外的其他算法,MS-本文IAFSA算法平均最优路径长度和平均运行时间均最短.而比D-ACO算法运行时间稍长的原因是MS-本文IAFSA算法要多处理 $K - 1$

条路径,因此在环境不是很复杂的情况下,为满足机器人导航的实时性要求, $K$ 值尽量不要取得过大,本实验中 $K = 3$ .

表2 8种算法100次重复实验结果对比

算法	最优路径长度/m	迭代次数	运行时间/s
MS-本文IAFSA	62.894 42	37	1.571 09
D-ACO <sup>[18]</sup>	65.000 73	39	1.378 42
GA <sup>[7]</sup>	102.948 15	72	14.461 07
fuzzy logic <sup>[11]</sup>	67.900 06	—	2.053 86
A* <sup>[2]</sup>	65.290 37	—	3.132 33
RRT <sup>[4]</sup>	86.314 14	—	8.851 45
PRM <sup>[5]</sup>	67.229 83	—	6.859 29
人工势场 <sup>[6]</sup>	153.028 02	—	6.918 74

### 3.5 算法复杂度及实用性分析

算法复杂度分析是指算法在编成可执行程序后运行时所需要的资源,通常包括时间复杂度分析与空间复杂度分析,人工鱼群算法属于群体算法,对于群体算法在实际应用中通常采用时间复杂度分析来估算算法执行效率的高低<sup>[21]</sup>.

通常将算法执行基本操作(如加、减、乘、除、比较等运算)的次数看作算法的时间复杂度.假设种群数量为 $N$ ,人工鱼的最大试探次数为try number,循环变量为 $t$ ,最大迭代次数为 $T_{max}$ ,则本文中IAFSA算法时间复杂度为

$$T_1(N) = O(t(3N^2 + \text{try number} \times N + 6N)). \quad (22)$$

MS算法的时间复杂度<sup>[16]</sup>为

$$T_2(m) = O(Km). \quad (23)$$

其中: $O$ 为数量级的概念, $K$ 为前最短路径数目, $m$ 为图中弧的数目.

MS-IAFSA总的时间复杂度为

$$T = T_1(N) + T_2(m). \quad (24)$$

本文算法的时间复杂度可以理解为执行算法编译的运行时间,由大量仿真时间和算法对比可知,本文改进算法可以求取高精度的全局最优解,并基本满足机器人导航的要求.

本文提出的MS-IAFSA主要用于移动机器人全局静态路径规划,若考虑实时路径规划,如机器人在按照规划好的路径运动时,发现自身内存存储的地图没有的障碍物,则机器人会首先通过自身的传感器采集障碍物信息,更新地图,然后重新进行路径规划.若遇到动态障碍物,则机器人通过自身传感器实时获取

障碍物的距离信息与速度信息,计算出障碍物的运动范围,然后将该区域标定为障碍区域,重新规划出一条避开障碍物区域的最优路径.

## 4 结论

本文在二维全局静态环境下,基于链接图法对移动机器人工作空间进行建模,并针对现有算法在移动机器人路径规划中的不足,提出了一种基于IAFSA算法和MAKLINK图的路径规划方法.引入指数递减惯性权重因子,使用Lorentzian函数和正态分布函数为视野和步长自适应算子,平衡了AFSA算法的全局与局部搜索能力,提高了AFSA算法的收敛速度与计算精度.提出MS算法与IAFSA算法分步寻优的策略,MS算法先计算前 $K$ 条最短路径,然后IAFSA算法进行二次优化,取二次优化后的最优路径为全局最优路径,解决了如ACO等算法在MAKLINK图只能求取近似最优路径的问题.最后进行了算法复杂度与实用性分析,大量仿真实验表明,所提出的基于IAFSA算法和MAKLINK图的路径规划方法能够有效地规划出全局最优路径.

本文是在二维全局静态环境下进行的研究,所有实验都是仿真实验,后续工作将在移动机器人本体上进行测试,并进行动态环境下的路径规划研究,以验证并进一步提高算法在真实环境下的路径规划能力.

### 参考文献(References)

- [1] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制与决策, 2010, 25(7): 961-967.  
(Zhu D Q, Yan M Z. Survey on technology of mobile robot path planning[J]. Control and Decision, 2010, 25(7): 961-967.)
- [2] 赵晓, 王铮, 黄程侃, 等. 基于改进A\*算法的移动机器人路径规划[J]. 机器人, 2018, 40(6): 903-910.  
(Zhao X, Wang Z, Huang C K, et al. Mobile robot path planning based on an improved A\* algorithm[J]. Robot, 2018, 40(6): 903-910.)
- [3] Zhang C F, Wang C L. Service robot path planning based on improved Dijkstra algorithm[C]. i-CREATE 2018 Proceedings of the 12th International Convention on Rehabilitation Engineering and Assistive Technology. Shanghai: Assistive and Rehabilitative Technologies (START) Centre, 2018: 77-80.
- [4] Kala R. Code for robot path planning using rapidly-exploring random trees[DB/OL]. (2014-04-10)[2018-12-30], <http://rkala.in/codes.html>.
- [5] Kala R. Code for Robot path planning using probabilistic roadmap[DB/OL]. (2014-04-10)[2018-12-

- 20]. <http://rkala.in/codes.html>.
- [6] Chen T B, Zhang Q S. Robot motion planning based on improved artificial potential field[C]. Proceedings of the 3rd International Conference on Computer Science and Network Technology. Dalian: IEEE, 2013: 1208-1211.
- [7] Ni J, Wang K, Huang H, et al. Robot path planning based on an improved genetic algorithm with variable length chromosome[C]. The 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). Changsha: IEEE, 2016: 145-149.
- [8] Liu J H, Yang J G, Liu H P, et al. An improved ant colony algorithm for robot path planning[J]. Soft Computing, 2017, 21(19): 5829-5839.
- [9] Tharwat A, Elhoseny M, Hassanien A, et al. Intelligent bézier curve-based path planning model using chaotic particle swarm optimization algorithm[J]. Cluster Computing, 2018: 1-22.
- [10] Hwu T, Wang A Y, Oros N, et al. Adaptive robot path planning using a spiking neuron algorithm with axonal delays[J]. IEEE Transactions on Cognitive and Developmental Systems, 2018, 10(2): 126-137.
- [11] Kala R. Code for robot path planning using fuzzy logic[DB/OL]. (2014-04-10)[2018-12-20]. <http://rkala.in/codes.html>.
- [12] 李晓磊. 一种新型的智能优化算法-人工鱼群算法[D]. 杭州: 浙江大学系统工程研究所, 2002.  
(Li X L. A new intelligent optimization method-artificial fish school algorithm[D]. Hangzhou: Institute of Systems Engineering, Zhejiang University, 2002.)
- [13] Zhang Y, Guan G L, Pu X C. The robot path planning based on improved artificial fish swarm algorithm[J]. Mathematical Problems in Engineering, 2016: 33(12): 374-385.
- [14] Huang Y Q, Wang P P, Yuan M R, et al. Path planning of mobile robots based on logarithmic function adaptive artificial fish swarm algorithm[C]. The 36th Chinese Control Conference (CCC). Dalian: IEEE, 2017: 4819-4823.
- [15] Chen T B, Zhang P, Gao S, et al. Research on the dynamic target distribution path planning in logistics system based on improved artificial potential field method-fish swarm algorithm[C]. 2018 Chinese Control and Decision Conference (CCDC). Shenyang: IEEE, 2018: 4388-4391.
- [16] Martins E Q V, Santos J L E. A new shortest paths ranking algorithm[J]. Investigacao Operacional, 2000, 20(1): 47-62.
- [17] 栾新源, 刘廷章, 周壮丽. 基于改进人工鱼群算法的LED混光方法[J]. 发光学报, 2015, 36(1): 113-120.  
(Luan X Y, Liu T Z, Zhou Z L. LED color mixing design based on improved artificial fish swarm algorithm[J]. Chinese Journal of Luminescence, 2015, 36(1): 113-120.)
- [18] 史峰, 王辉, 郁磊, 等. Matlab智能算法30个案例分析[M]. 北京: 北京航空航天大学出版社, 2011: 217-227.  
(Shi F, Wang H, You L, et al. Analysis of 30 cases of intelligent algorithms in Matlab[M]. Beijing: Beihang University Press, 2011: 217-227.)
- [19] Habib M K, Asama H. Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach[C]. Proceedings IROS 91: IEEE/RSJ International Workshop on Intelligent Robots and Systems. Osaka: IEEE, 1991: 563-567.
- [20] 李明富, 张玉彦, 马建华, 等. 基于变参数萤火虫算法和Maklink图的路径规划研究[J]. 机械科学与技术, 2015, 34(11): 1728-1732.  
(Li M F, Zhang Y Y, Ma J H, et al. Research on path planning based on variable parameters firefly algorithm and maklink graph[J]. Mechanical Science and Technology for Aerospace Engineering, 2015, 34(11): 1728-1732.)
- [21] Manber Udi. Introduction to algorithms—A creative approach[M]. Vancouver: Addison-Wesley Publishing Company, 1989: 37-43.

### 作者简介

郭伟(1965—), 男, 教授, 博士, 从事现代设计理论方法与技术、人工智能、大数据等研究, E-mail: wguo@tju.edu.cn;

秦国选(1994—), 男, 硕士生, 从事移动机器人的路径规划与导航的研究, E-mail: qin1021@tju.edu.cn;

王磊(1976—), 男, 副教授, 从事现代设计理论方法与技术、人工智能等研究, E-mail: w2165@139.com;

孙日杰(1992—), 男, 硕士生, 从事移动机器人视觉跟随的研究, E-mail: srjie@tju.edu.cn.

(责任编辑: 郑晓蕾)