

DOI: 10.11992/tis.201606046

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.TP.20170404.1218.004.html>

基于改进粒子群算法的移动机器人多目标点路径规划

蒲兴成¹, 李俊杰², 吴慧超², 张毅³

(1.重庆邮电大学 数理学院, 重庆 400065; 2.重庆邮电大学 智能系统及机器人研究所, 重庆 400065; 3.重庆邮电大学 先进制造学院, 重庆 400065)

摘要: 针对移动机器人遍历多个目标点的路径规划问题, 提出了一种基于改进粒子群算法和蚁群算法相结合的路径规划新方法。该方法将目标点的选择转化为旅行商问题, 并利用蚁群算法进行优化, 定义了每两个目标点之间的路径规划目标函数, 利用粒子群算法对其进行优化。针对粒子群算法存在的早熟现象, 将反向学习策略引入粒子群算法, 并对粒子群算法的惯性权重和学习因子进行改进。性能测试结果表明, 改进的粒子群算法能有效避免粒子早熟现象, 提高粒子群算法的寻优能力及稳定性。仿真实验结果验证了新方法能有效地实现机器人的多目标点无碰撞路径规划。真实环境下的实验结果证明了新方法在机器人多目标点路径规划的实际应用中也具有有效性。

关键词: 移动机器人; 多目标点路径规划; 蚁群算法; 改进粒子群算法; 反向学习策略; 惯性权重; 学习因子

中图分类号: TP242.6 **文献标志码:** A **文章编号:** 1673-4785(2017)03-0301-09

中文引用格式: 蒲兴成, 李俊杰, 吴慧超, 等. 基于改进粒子群算法的移动机器人多目标点路径规划[J]. 智能系统学报, 2017, 12(3): 301-309.

英文引用格式: PU Xingcheng, LI Junjie, WU Huichao, et al. Mobile robot multi-goal path planning using improved particle swarm optimization[J]. CAAI transactions on intelligent systems, 2017, 12(3): 301-309.

Mobile robot multi-goal path planning using improved particle swarm optimization

PU Xingcheng¹, LI Junjie², WU Huichao², ZHANG Yi³

(1.School of Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 2.Research Center of Intelligent System and Robot, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 3.Advanced Manufacturing Engineering School, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: To solve the problem of multi-goal path planning for mobile robots that pass multiple goals, a new path planning method, based on improved particle swarm optimization (PSO) and ant colony optimization (ACO), is proposed. In this new method, the first step is to use an improved PSO, which has high convergence, to optimize the path between two goals among a sequence of goals. The second step is to use the ACO to obtain the shortest path for all target points. The performance experimental result demonstrates that the improved PSO algorithm can effectively avoid premature convergence and enhances search ability and stability. Simulation results show that the improved PSO algorithm can make a mobile robot realize collision-free multi-goal path planning effectively. Experiments in a real environment demonstrate that this algorithm has practical application for multi-goal path planning.

Keywords: mobile robot; multi-goal path planning; ACO; improved PSO; opposition-based learning; inertia weight; learning factors

路径规划是研究移动机器人的一个重要内容,

按其规划范围,分为全局路径规划和局部路径规划。目前,针对这两种路径规划方式许多学者进行了深入研究,并提出了相应的解决方法。全局路径规划方法有栅格法、可视图法和神经网络法等;局

收稿日期: 2016-06-30. 网络出版日期: 2017-04-04.

基金项目: 国家自然科学基金(51604056), 重庆市科学技术委员会项目(cstc2015jcyBx0066); 重庆市教委项目(KJ1400432).

通信作者: 李俊杰, E-mail: lijunjie166@126.com.

部路径规划方法包括人工势场法、 A^* 算法、人工蜂群算法等^[1]。虽然这些算法得到广泛应用,但也存在一些缺点:栅格力度越小,栅格法对障碍物的描述越精确,但环境信息的存储会占据大量的存储空间,算法的搜索范围也将成指数增加^[2];人工势场法由于斥力的作用,机器人很难准确到达目标点。 A^* 算法对开放列表的维护是最耗时的,尤其是在地图过大的情况下。

移动机器人多目标点路径规划是指在复杂的环境下找到一条从起始点开始经过所有目标点的无碰撞的最优路径。在实际应用中,移动机器人多目标点路径规划多应用于电厂巡检、医院查房等。移动机器人多目标点路径规划的基本原则包括3个方面:1)当前点或起始点以及目标点提前已知;2)任意时刻机器人都可以决定移动路线;3)机器人必须经过的所有目标点来完成路径规划。根据移动机器人多目标点路径规划的特点,移动机器人多目标点路径规划问题可以转化为旅行商(traveling salesman problem, TSP)问题。目前有很多算法用于解决 TSP 问题^[3],但多目标点路径规划与 TSP 问题相比更复杂。因此本文提出了一种基于粒子群算法和蚁群算法相结合的混合算法用于解决移动机器人多目标点路径规划问题。

蚁群算法(ant colony optimization, ACO)是一种启发式进化算法,由蚂蚁觅食行为演化而来。在解决 TSP 问题上,蚁群算法表现出了较强的优越性。粒子群算法(particle swarm optimization, PSO)是一种群体智能算法,由鸟群觅食行为演化而来^[4]。与一般的群体智能算法相比,PSO 具有记忆特性,可以通过自我学习和向他人学习方式获得更多的信息。由于参数少,计算方便等优点,PSO 广泛应用于优化问题上,并取得了很好的效果。随着移动机器人技术的发展,国内外学者对粒子群算法在移动机器人路径规划上做了大量的研究,如多机器人路径规划^[5-6],自由浮动机器人轨迹规划^[7]以及其他应用领域^[8-9]。PSO 在路径规划上应用广泛,ACO 善于解决 TSP 问题,因此采用两者结合的方式非常适合移动机器人多目标点路径规划问题。

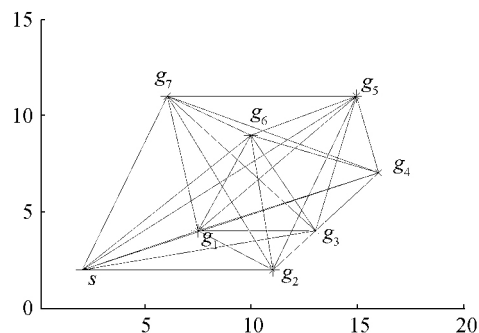
在一个包含障碍物的空间内,利用 PSO 算法对起始点到目标点及每个目标点与目标点之间进行路径规划。起始点、目标点以及障碍物位置已知,但是哪一条路线会被选择是未知的。为了确定哪一条路径会被选择,ACO 被用于遍历起始点和所有目标点,以便得到一条经过所有点的最短路径。为

了避免 PSO 出现早熟现象,提高算法的效率,提出了具有快速收敛性的粒子群算法。实验结果验证了混合算法的有效性以及改进的粒子群算法的优越性。

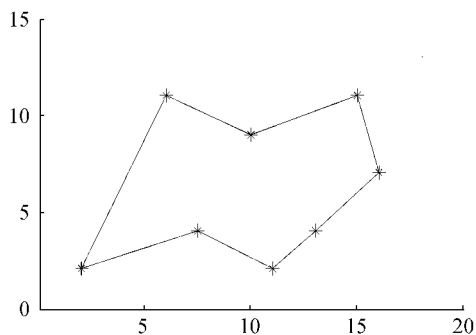
1 问题描述

1.1 多目标点路径规划

假设在移动机器人多目标点路径规划中含有 n 个目标点,则移动机器人会产生 $n(n-1)/2$ 条路径。为了满足移动机器人行走路径最短的要求,机器人需要从 $n(n-1)/2$ 条路线中选择出一条经过所有点的最短路径。图1描述了在起始点和目标点位置已知的情况下,移动机器人的运动轨迹。从图1(a)中可知移动机器人的移动路线有很多条,且必须经过所有目标点才能完成路径规划;图1(b)表示移动机器人运动的理想路线。



(a) 所有路径



(b) 理想路径

图1 多点之间的路径轨迹

Fig.1 The path trajectory of multi-goal

为了解决上述问题,我们将目标点的选择规划为 TSP 问题的一个分支。目前,求解 TSP 问题最有效的方法主要有郭涛算法、模拟退火算法、遗传算法、蚁群算法等^[3]。由于这些算法都有优缺点,因此有些学者尝试结合两种或多种算法解决彼此算法上的缺点^[10]。蚁群算法作为一种自组织正反馈算法,与其他算法相比,具有很强的鲁棒性。而且

从本质上讲,蚁群算法的并行特性,强化了算法的可靠性。因此在解决多目标点选择的问题上,我们将采用蚁群算法搜索能够遍历所有目标点的方法。

1.2 粒子的适应度函数

粒子的适应度函数是检验粒子群算法收敛性的重要函数。在路径规划中,移动距离最短是机器人首要考虑的目标,其次是安全性。因此,将移动机器人运动距离作为粒子的首要适应度函数。移动机器人多目标点路径规划如图2所示。由图2可知,移动机器人从起始位置开始避开障碍物,经过一系列目标点到达目标点 n 。

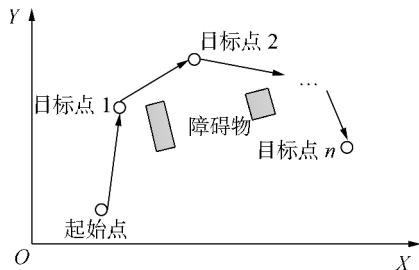


图2 移动机器人多目标点路径规划示意图

Fig.2 The diagram of mobile robot multi-goal path planning

为了便于理解移动机器人多目标点路径规划,可将图2中的多目标点路径规划分解为多个目标点中的两两目标点之间路径规划。机器人在保证两点之间路径规划最短的基础上才能保证多目标路径规划最短。分解后如图3所示,机器人在当前点需要选择下一个所经过点,然后到达目标点,在此期间要避开障碍物。

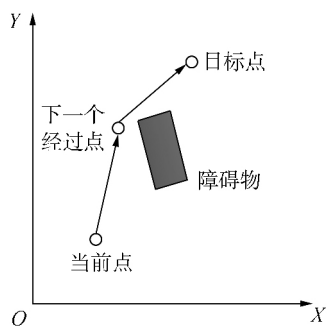


图3 当前点到目标点避障描述

Fig.3 The obstacle avoidance description of current point to goal

综上所述,为了保证路径规划的最短性,下一个移动点的选取是保证移动机器人运动轨迹最短的前提。因此,设立移动机器人路径规划目标函数 F_1 , F_1 决定了移动机器人行走路径长度。本文中, F_1 定义为

$$F_1 = \sum_{j=1}^g \sum_{i=1}^S \left[\sqrt{(x_{ij}^{\text{current}} - x_{ij}^{\text{next}})^2 + (y_{ij}^{\text{current}} - y_{ij}^{\text{next}})^2} + \sqrt{(x_{ij}^{\text{next}} - x_{ij}^{\text{goal}})^2 + (y_{ij}^{\text{next}} - y_{ij}^{\text{goal}})^2} \right]$$

式中 g 是目标点的个数, S 是粒子个数。

移动机器人在向目标点移动过程中,除了要保证其最短性,安全性也是必须要考虑的。为此,将障碍物对目标点的斥力场函数作为安全性的惩罚函数。斥力场函数定义如下:

$$F_2 = \begin{cases} k_r \times \left(\frac{1}{d(x_R, x_O)} - \frac{1}{d_m} \right)^2 d^2(x_R, x_G), & d(x_R, x_O) \leq d_m \\ 0, & d(x_R, x_O) > d_m \end{cases}$$

式中: F_2 是障碍物对机器人的斥力; k_r 是斥力场系数; $d(x_R, x_O)$ 为机器人到障碍物的距离; $d(x_R, x_G)$ 是机器人到目标点的距离; d_m 是障碍物的影响范围。

移动机器人多目标点路径规划问题可以抽象为当前位置到目标点的最短距离的优化问题,而障碍物对机器人的斥力作用是保证最短路径安全性的前提。因此,针对路径规划的目标函数可以定义为

$$F = \lambda F_1 + \mu F_2$$

式中 λ 和 μ 分别是最短路径和约束条件的权重。

2 基本知识

2.1 标准粒子群算法

在PSO算法中,每个粒子通过自我学习及向“他人”学习向问题空间中的最佳位置移动,进而找到最优解。 N 维空间内粒子的位置变量为 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$ 为粒子的速度变量。粒子迭代时,通过向两个“极值”学习来更新自己:1)局部最优值 p_{best} ;2)全局最优值 g_{best} 。PSO充分利用局部最优值和全局最优值,可使优秀粒子的基因得到传承。粒子寻优的过程是粒子根据自我学习和向他人学习来更新自己的速度和位置,逐步获得最优解。粒子的速度和位置更新公式为

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_{\text{best}} - x_i^t) + c_2 r_2 (g_{\text{best}} - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (2)$$

式中: ω 是粒子的惯性权重; c_1 、 c_2 是学习因子; r_1 、 r_2 是 $[0, 1]$ 的随机数; x_i^t 表示粒子 t 的位置且 $x_i^t \in [x_{\min}, x_{\max}]$;速度 $v_i^t \in [v_{\min}, v_{\max}]$ 。当粒子的位置超出了 $[x_{\min}, x_{\max}]$ 时,粒子被限制为 x_{\min} 或 x_{\max} ;若粒子的速度超出了 v_{\max} 时,粒子的速度被限制为最大速度 v_{\max} 。

2.2 反向学习策略

Tizhoosh^[11]在2005年基于相反数或对立点的概念提出了反向学习策略。在随后的发展中,该方法被应用于解决一些优化问题中^[12-13]。反向坐标定义如下。

定义1^[14] 假设 x 在一维空间中的坐标是 -5 ,目标点的坐标为 10 。那么 x 反向点 \tilde{x} 的坐标点为 5 ,且点 x 到目标点的距离为 15 ,点 \tilde{x} 到目标点的距离为 5 。比较可得,反向点靠近目标点。 x 的反向坐标点 \tilde{x} 的计算公式为

$$\tilde{x} = a + b - x$$

式中 $x \in [a, b]$,且为实数。把上述定义应用到高维空间可表示成定义2。

定义2^[14] 在 N 维空间中的一个点 P 可表示为 $P(x_1, x_2, \dots, x_n)$,其中 $x_1, x_2, \dots, x_n \in \mathbf{R}$,且 $x_i \in [a_i, b_i], \forall i = \{1, 2, \dots, n\}$ 。则点 P 对应的反向点 \tilde{P} 可表示为 $\tilde{P}(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$,其中 \tilde{x}_i 表示为

$$\tilde{x}_i = a_i + b_i - x_i$$

将定义2应用到优化过程中,则反向机制的优化过程如定义3。

定义3^[14] 根据定义1和定义2,反向学习可以描述为:假设 $f(x)$ 是已知问题的函数方程, n 维空间中的点 $P_i(x_1, x_2, \dots, x_n)$ 是函数方程 $f(x)$ 的一个解,且 $x_i \in [a_i, b_i], \forall i = \{1, 2, \dots, n\}$ 。则 P_i 的反向点为 $\tilde{P}_i(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ 。计算 P_i 和 \tilde{P}_i 在函数 $f(x)$ 中的函数值。如果 $f(\tilde{P}_i) > f(P_i)$,则反向点 \tilde{P}_i 代替 P_i ;否则保留 P_i 。因此通过不断地计算点 P_i 与反向点 \tilde{P}_i 的 $f(x)$ 值选择其中最优值。

3 反向学习粒子群算法

基于反向学习策略的思想,本文提出一种改进的粒子群算法(opposition-based learning improved PSO, OBLPSO)。本文采用反向学习策略时,将反向学习策略应用到单个粒子中,而不是粒子群体中,以便减小粒子迭代过程中的计算量。性能测试结果证明,改进的粒子群算法能抑制粒子早熟现象,提高收敛效率。下面介绍改进算法的主要机制和流程。

3.1 初始化种群

粒子在问题空间中找到最优解的时间与粒子初始化时在空间中的分布存在正比关系,距离最佳位置近的粒子找到最优解的时间比距离最佳位置

远的粒子要快。但是,粒子在问题空间中是随机分布的,每一个粒子相对于最优解的位置都是未知的。

基于以上分析,在改进的粒子群算法初始化时引入反向学习策略有助于粒子寻找最优解。在初始化时,首先,计算粒子的适应度值以及其反向粒子适应度值,并对两者进行比较,选择适应度值较好的粒子;其次,从种群中选择 S 个适应度值最好的粒子作为初始种群。

3.2 迭代进化过程

在标准PSO算法中,粒子在问题空间中选择另一个粒子作为学习对象,根据式(1)、(2)进行移动。但在粒子进化的过程中,粒子会发生早熟现象,导致算法无法得到最优解。由式(1)可知,粒子的速度受到惯性权重和学习因子的影响:惯性权重影响着粒子的全局搜索和局部搜索能力,学习因子影响着粒子获取信息的能力。为了找到问题最优解,粒子在进化的过程中,其全局搜索能力和局部搜索能力也应该随之发生变化,即从全局搜索渐变为局部搜索,保证粒子可以寻找到问题的最优解。同样,粒子也应该逐渐增强社交能力,由“自我”学习逐渐向“他人”学习过渡,以便获取更多有用的信息。

根据以上分析,粒子在寻优过程中,惯性权重应该保持动态变化。 ω 取值较大时,粒子的全局搜索能力较强; ω 取值较小时,粒子的局部搜索能力较强。由于粒子在寻优的过程中,会逐渐靠近最优解,粒子的惯性权重应该随着寻优过程自适应变化。因此粒子的惯性权重更新公式为

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \left(1 - \frac{\text{dist}_i}{\max_dist} \right)$$

式中: ω_{\max} 和 ω_{\min} 分别是粒子惯性权重的最大值和最小值; dist_i 是第 i 个粒子到全局最优粒子的欧几里德距离; \max_dist 是粒子到全局最优粒子的最大距离。 dist_i 和 \max_dist 表达式分别为^[15]

$$\begin{aligned} \text{dist}_i &= \left(\sum_{d=1}^D g_{\text{best}_d} - x_{i,d} \right)^{\frac{1}{2}} \\ \max_dist &= \arg\max_i (\text{dist}_i) \end{aligned}$$

学习因子 c_1 和 c_2 控制粒子本身记忆与同伴记忆之间的相对影响: c_1 的值较小时,表现为自我认知能力不足; c_2 的值较小时,表现为社交能力不足。为了保证在迭代过程中,粒子的自我认知和社交能力能够因时而异,因此,粒子的学习因子更新公式^[16]如下:

$$c_1 = c_{1\max} \times \left(\frac{c_{1\min}}{c_{1\max}} \right)^{\left(\frac{t}{t_{\max}} \right)}$$

$$c_2 = c_{2\min} \times \left(\frac{c_{2\max}}{c_{2\min}} \right)^{\left(\frac{t}{T_{\max}} \right)}$$

式中: $c_{1\max}$ 和 $c_{2\max}$ 以及 $c_{1\min}$ 和 $c_{2\min}$ 分别是学习因子 c_1 和 c_2 的最大值和最小值; t 是当前的迭代次数; T_{\max} 是最大迭代次数。

3.3 算法实现流程

1) 首先初始化种群 $P(S)$, 包括粒子的位置 x_i 和速度 v_i , $i=1, 2, \dots, S$, S 是种群的大小, 初始化惯性权重 ω 以及学习因子 c_1 和 c_2 等参数。

2) 在速度和位置规定的范围内根据式(1)、(2)更新第 i 个粒子的速度 v_i 和位置 x_i 。

3) 计算第 i 个粒子的适应度值 f_i 。

4) $a_i = \min(x_i)$ 。

5) $b_i = \max(x_i)$ 。

6) 计算第 i 个粒子的位置和速度反向解: $\tilde{x}_i = a_i + b_i - x_i$ 。

7) 计算第 i 个粒子反向解的适应度值 \tilde{f}_i 。

8) 比较 f_i 和 \tilde{f}_i 的大小, 如果 $\tilde{f}_i < f_i$, 则 $x_i = \tilde{x}_i$, $f_i = \tilde{f}_i$ 。

9) 从最优适应度值中选出 S 个适应度值最好的粒子组成初始化种群。

10) 接下来同粒子群算法基本流程。

11) 算法满足结束条件, 结束算法过程。

4 实验结果与分析

4.1 OBLIPSO 性能测试

为了验证改进粒子群算法的性能是否得到改进, 笔者将反向学习的粒子群算法与其他改进算法^[15, 17-18]在4个适应度函数上进行对比。这4个适应度函数分别为 Sphere 函数、Rastrigin 函数、Grewank 函数、Schaffer 函数, 测试函数参数设置如表1所示。

表1 测试函数参数设置

Table 1 The parameter setting of test function

函数	取值范围	最优点	最优解
Sphere(f_1)	[-100, 100]	(0, 0)	0
Rastrigin(f_2)	[-5.12, 5.12]	(0, 0)	0
Grewank(f_3)	[-600, 600]	(0, 0)	0
Schaffer(f_4)	[-100, 100]	(0, 0)	0

OBLIPSO 算法参数设置为: 粒子群大小为 40, 空间维度为 2, $\omega = [0.4, 0.9]$, $c_1 = [1.25, 2.75]$, $c_2 = [0.5, 2.25]$, 最大迭代次数为 100。

在表2中, OBLIPSO 算法在四测试函数上的平均值均小于其他3种算法的值, 尤其是在 Sphere 函数和 Rastrigin 函数中。从表2中数据可知, OBLIPSO 算法具有更好的收敛性, 得到的解更优。表3中的数据反映出改进算法的稳定性。OBLIPSO 算法与 IAPSO 算法、IPSO 算法和 WPSO 算法对比, 在稳定性上表现得更好。

表2 平均最优值结果对比

Table 2 The comparison of average best values

函数	OBLIPSO	IAPSO	IPSO	WPSO
f_1	0	0.132 790	0.020 476	0.034 939
f_2	0	0.066 379	0.066 331	0.298 488
f_3	0.002 153	0.031 692	0.017 342	0.018 080
f_4	0.003 273	0.005 328	0.006 179	0.017 561

表3 标准方差结果对比

Table 3 The comparison of standard deviation

函数	OBLIPSO	IAPSO	IPSO	WPSO
f_1	0	0.714 775	0.009 423	0.006 733
f_2	0	0.248 174	0.248 186	0.455 948
f_3	0.004 418	0.087 505	0.021 595	0.019 151
f_4	0.004 560 0	0.004 699	0.004 650	0.032 801

由表2和表3数据可得: 反向学习策略提高了种群的多样性, 增加了粒子寻优成功概率, 节省了粒子寻优时间; 线性变化的惯性权重保障了粒子从全局搜索到局部搜索的线性转换, 抑制了粒子早熟现象; 学习因子的变化保证了粒子能够充分完成自我学习和社交行为, 提高了粒子的收敛速度。

4.2 路径规划仿真实验

为了验证新方法的实用性及有效性, 笔者将该方法在仿真环境下进行实验。仿真实验环境设置在 18×16 的二维矩形空间内, 并设置成简单环境和复杂环境。在环境中有不规则的障碍物, 起始点位置使用方形表示, 目标点处用五角形表示。

1) 第1次实验中, OBLIPSO 算法的参数设置为: $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, $S = 10$, $c_{1\max} = 2.75$, $c_{2\max} = 2.25$, $c_{1\min} = 1.25$, $c_{2\min} = 0.5$, $v_{\max} = 1.9$, $v_{\min} = 0$, 斥力场中的安全距离 $d_m = 2$, $k_r = 2.7$, $\lambda = 0.25$, $\mu = 0.25$, 最大迭代次数为 100。图4(a)给出了在简单环境下的 OBLIPSO 作用的移动机器人多目标点移动轨迹。图4(b)是 IAPSO 作用的运动轨迹。图4(c)是 IPSO 作用的运动轨迹。

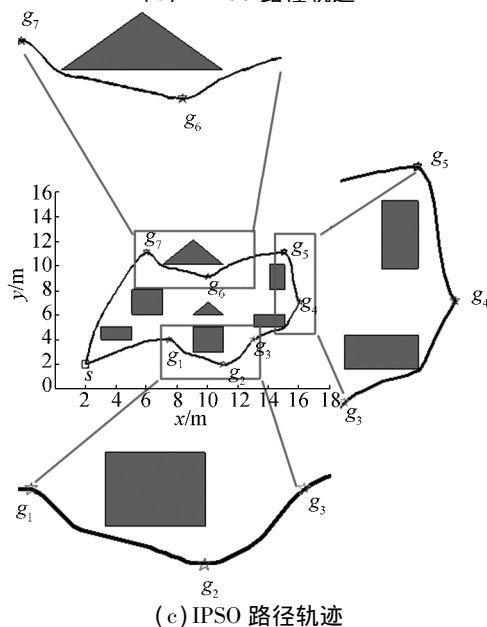
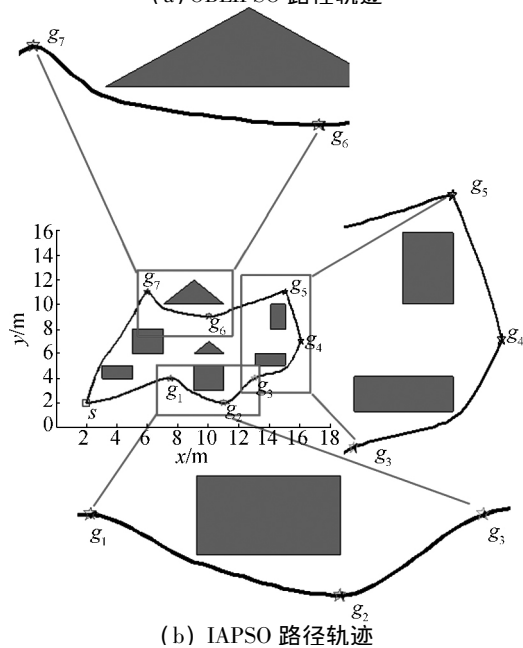
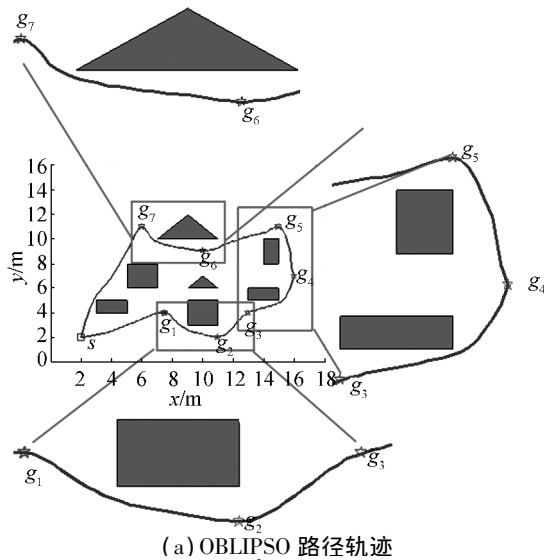
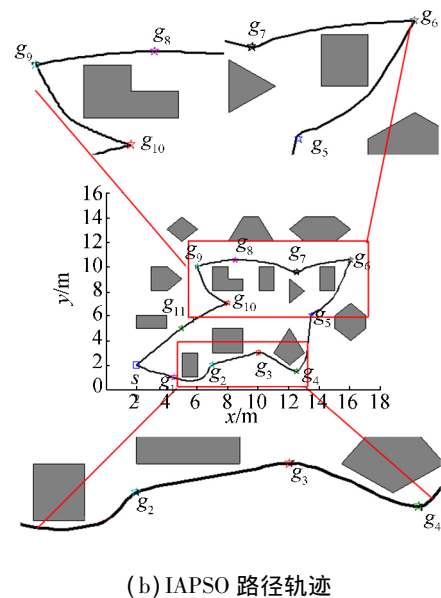
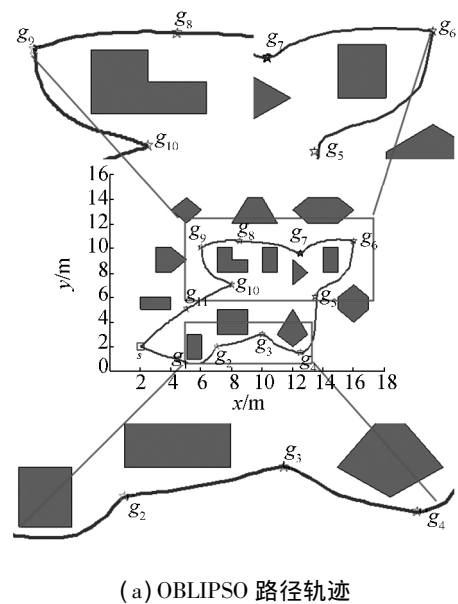


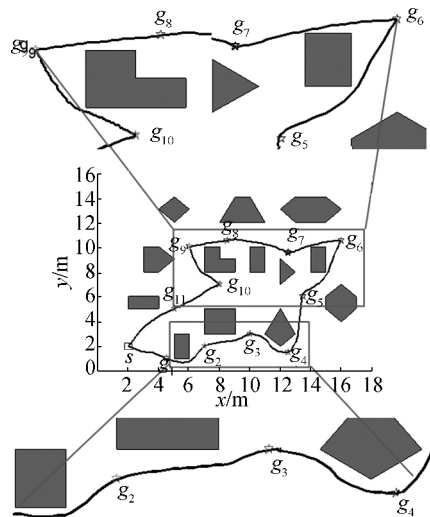
图4 简单环境下的优化过程

Fig.4 The motion process in the first environment

由图4框中路径轨迹可知:在目标点3到目标点4和目标点6到目标点7之间,OBLIPSO和IAPSO的避障性能上优于IPSO,在经过目标点4和目标点5时,OBLIPSO的路径较为平滑。而且OBLIPSO在迭代72次左右时成功完成路径规划。

2)在第2次实验中,实验环境相对第1次实验而言,增加了障碍物以及目标点的个数。实验参数设置如下: $v_{\max} = 20$,最大迭代次数为300,其余参数与第1次实验中参数相同。由于环境复杂度的增加,相对第1次实验而言,粒子收敛速度变慢。图5(a)是在复杂环境下的OBLIPSO获得的移动机器人多目标点运动轨迹。图5(b)是IAPSO获得的运动轨迹。图5(c)是IPSO作用的运动轨迹。





(c) IPSO 路径轨迹

图 5 复杂环境下迭代过程

Fig.5 The iteration process in complicated environment

对比图 5 中 3 条轨迹可知,随着障碍物和目标点增加,3 种算法都满足了路径的可达性。根据图 5 框中的路径轨迹可知,在经过目标点 6 和目标点 9 时,OBLIPSO 算法规划的路径相比于 IAPSO 和 IPSO 算法规划的路径更为平滑。OBLIPSO 算法规划的路径在目标点 5 与目标点 10 之间相比于 IAPSO 和 IPSO 算法规划的路径安全性更好,OBLIPSO 移动距离较短(见表 4),而且其规划的路径安全性最好。

3) 路径规划性能对比

重复 10 次上述两种任务的实验,取其平均值。移动机器人多目标点路径规划的时间消耗和移动距离性能对比如表 4 所示。

表 4 路径性能对比

Table 4 The performance comparison of path

环境	算法	移动距离/m	消耗时间/s
简单环境	OBLIPSO	40.943	8.866
	IAPSO	41.749	10.257
	IPSO	42.865	10.625
复杂环境	OBLIPSO	44.942	18.583
	IAPSO	43.127	23.265
	IPSO	45.336	25.186

从表 4 数据中可以得出,随着任务复杂度的增加,OBLIPSO、IAPSO 和 IPSO 在时间消耗上也在增加,任务越复杂,用时越多。但在进行相同任务时,OBLIPSO 用时比 IAPSO 和 IPSO 少;在移动距离方面,简单任务时,OBLIPSO 的移动距离比其他两种对比算法短,在复杂任务时,虽然 OBLIPSO 的移动

距离长于 IAPSO,但其比 IPSO 算法短;在路径安全性方面,OBLIPSO 算法要优于其他两种对比算法。由于在设计路径规划目标函数时不仅考虑了路径最短性问题,也考虑到了其安全性的问题,因此综合两种环境下实验结果可以得出 OBLIPSO 算法综合性能强于 IAPSO 算法和 IPSO 算法。

图 6 给出的是简单环境下 OBLIPSO 算法、IAPSO 算法和 IPSO 算法在目标函数上的收敛曲线对比。从图 6 中曲线可以得出,在迭代 72 次左右时,OBLIPSO 算法可在多目标点路径规划中找到最优路径,而 IAPSO 算法和 IPSO 算法在迭代 100 次时仍未获得 OBLIPSO 算法的收敛值。

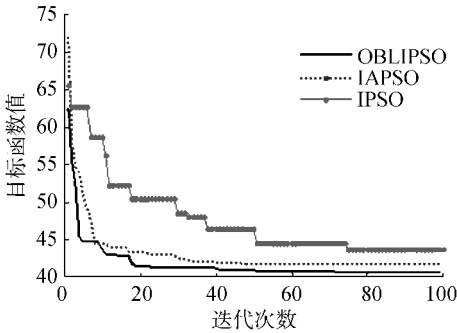


图 6 OBLIPSO 与 IAPSO 和 IPSO 目标函数收敛曲线比较

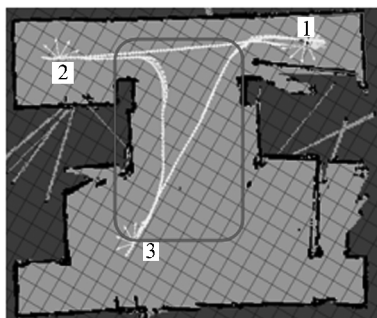
Fig.6 The convergence comparison of OBLIPSO with IAPSO and IPSO

综上所述,在相同任务下,OBLIPSO 算法的综合性能优于 IAPSO 算法和 IPSO 算法。随着任务复杂度的变化,OBLIPSO 算法能够有效且较好地完成移动距离和时间消耗以及安全性上表现得更好,表明了 OBLIPSO 算法能够有效且较好地完成移动机器人多目标点路径规划上有效性。

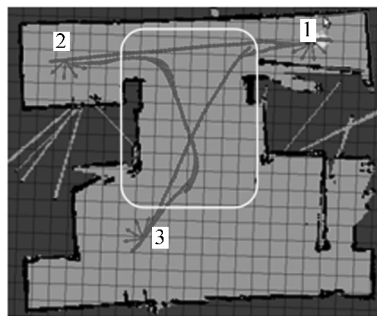
4.3 实际环境下路径规划

为了验证 OBLIPSO 算法在实际应用中的有效性,笔者将在实际环境下对 OBLIPSO 算法进行实验验证。实验环境设置为重庆邮电大学自动化学院办公楼一楼。实验平台为先锋 3 机器人,软件背景为 ROS(robot operation system)^[19-21]。实验中设置了 3 个目标点,机器人从目标点 1 出发经过两个目标点,最后返回目标点 1。

图 7 中显示的是机器人在无障碍环境中完成的多目标点路径规划。图 7(a)为 OBLIPSO 算法规划的路径,图 7(b)为 IAPSO 算法规划的路径。由图 7 可知:机器人从目标点 2 到目标点 3 的过程中,OBLIPSO 算法能够保证机器人与墙体保持安全距离且路径平滑;IAPSO 算法虽然保证了机器人与墙的安全距离,但由于机器人远离墙体移动造成了路径的交叉,如图 7 中方框所示。



(a) OBLIPSO 路径轨迹



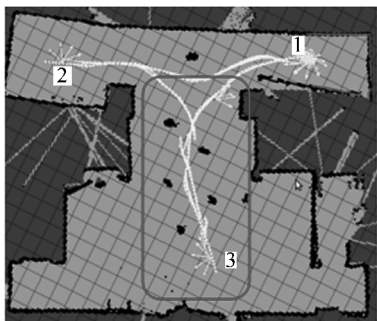
(b) IAPSO 路径轨迹

图7 机器人在无障碍环境下的移动过程

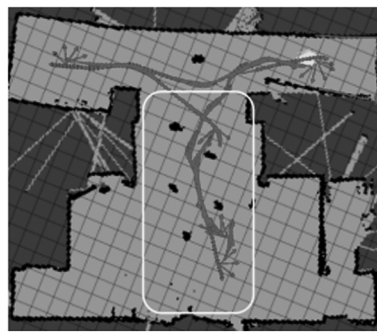
Fig.7 The motion process of robot in barrier-free environment

为了进一步验证 OBLIPSO 算法的有效性,环境中设置了障碍物,并改变了目标点3的位置,如图8所示。图8表示机器人在含有障碍物的环境下实现的多目标点路径规划。图8(a)为 OBLIPSO 算法规划的路径,图8(b)为 IAPSO 算法规划的路径。由图8可知:在含有障碍物的环境中,机器人在两种算法下都能够避开障碍物完成路径规划,但是机器人从目标点2到目标点3的过程中,OBLIPSO 算法能够保证机器人与障碍物保持安全距离并且路径平滑;IAPSO 算法在移动过程中,虽然机器人与障碍物保持了安全的距离,但由于其路径的交叉导致移动轨迹较长,如图8中方框所示。

表5为两次实验重复10次取得的时间和路径长度的平均值。由表5可知:在障碍环境下,OBLIPSO 算法可以实现机器人多目标点的路径规划,并且消耗的时间较少。



(a) OBLIPSO 路径轨迹



(b) IAPSO 路径轨迹

图8 机器人在障碍环境下的移动过程

Fig.8 The motion process of robot in the obstacle environment

表5 真实环境下路径性能对比

Table 5 The performance comparison of path in real environment

指标	无障碍	有障碍
	(OBLIPSO/IAPSO)	(OBLIPSO/IAPSO)
时间/s	114/114	186/216
距离/m	20.1/20.1	19.5/20

5 结束语

本文提出了一种结合 ACO 算法和 OBLIPSO 算法的移动机器人多目标点路径规划新方法,并通过实验验证了新方法的可行性。ACO 算法在 TSP 问题上表现出较好的优越性,因此 ACO 算法适用于移动机器人的目标点选择问题。OBLIPSO 算法不仅继承了 PSO 算法计算简单的特点,而且在初始化时引入反向学习策略,保证了粒子的多样性。在迭代过程中,OBLIPSO 算法抑制了粒子的早熟,提高了收敛速度。采用4个适应度函数对 OBLIPSO 算法进行性能评估,取得了良好的测试结果。仿真实验与真实环境下实验结果表明提出的新方法是一种有效的多目标点路径规划方法。将本文提出的新方法应用到动态环境下的移动机器人的多目标点路径规划是下一步的研究内容。

参考文献:

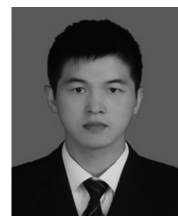
- [1]杨兴,张亚,杨巍,等.室内移动机器人路径规划研究[J].科学技术与工程,2016,16(15):234-238.
YANG Xing, ZHANG Ya, YANG Wei, et al. Research on path planning of indoor mobile robot [J]. Science technology and engineering, 2016, 16 (15): 234-238.
- [2]Ammar A, Bennaceur H, Châari I, et al. Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments [J]. Soft computing, 2016, 20 (10):4149-4171.
- [3]杜鹏桢,唐振民,孙研.一种面向对象的多角色蚁群算法及其 TSP 问题求解[J].控制与决策,2014(10):1729-1736.
DU Pengzhen, TANG Zhenmin, SUN Yan. An object-oriented multi-role ant colony optimization algorithm for solving

- TSP problem [J]. Control and decision, 2014 (10): 1729-1736.
- [4] AGRAWAL R K, BAWANE N G. Multiobjective PSO based adaption of neural network topology for pixel classification in satellite imagery [J]. Applied soft computing, 2015, 28 (C): 217-225.
- [5] DAS P K, BEHERA H S, PANIGRAHI B K. Intelligent-based multi-robot path planning inspired by improved classical Q-learning and improved particle swarm optimization with perturbed velocity [J]. Engineering science and technology, an international journal, 2015, 19(1): 651-669.
- [6] YANG Mao, LI Chunzhe. Path planning and tracking for multi-robot system based on improved PSO algorithm [C]//Proceedings of 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC). Jilin: IEEE, 2011: 1667-1670.
- [7] WANG Mingming, LUO Jianjun, WALTER U. Trajectory planning of free-floating space robot using particle swarm optimization (PSO) [J]. Acta astronautica, 2015, 112: 77-88.
- [8] 张万绪, 张向兰, 李莹. 基于改进粒子群算法的智能机器人路径规划 [J]. 计算机应用, 2014, 34 (2): 510-513.
ZHANG Wanxu, ZHANG Xianglan, LI Ying. Path planning for intelligent robots based on improved particle swarm optimization algorithm [J]. Journal of computer applications, 2014, 34(2): 510-513.
- [9] 王娟, 吴宪祥, 郭宝龙. 基于改进粒子群优化算法的移动机器人路径规划 [J]. 计算机工程与应用, 2012, 48 (15): 240-244.
WANG Juan, WU Xianxiang, GUO Baolong. Robot path planning using improved particle swarm optimization [J]. Computer engineering and applications, 2012, 48 (15): 240-244.
- [10] 张勇, 陈玲, 徐小龙, 等. 基于 PSO-GA 混合算法时间优化的旅行商问题研究 [J]. 计算机应用研究, 2015, 32(12): 3613-3617.
ZHANG Yong, CHEN Ling, XU Xiaolong, et al. Research on time optimal TSP based on hybrid PSO-GA [J]. Application research of computers, 2015, 32(12): 3613-3617.
- [11] TIZHOOSH H R. Opposition-based learning: a new scheme for machine intelligence [C]//Proceedings of 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on Computational Intelligence for Modelling, Control and Automation. Vienna: IEEE, 2005: 695-701.
- [12] MUÑOZ D M, LLANOS C H, COELHO L D S, et al. Hardware opposition-based PSO applied to mobile robot controllers [J]. Engineering applications of artificial intelligence, 2014, 28: 64-77.
- [13] 汪慎文, 丁立新, 谢大同, 等. 应用反向学习策略的群搜索优化算法 [J]. 计算机科学, 2012, 39(9): 183-187.
WANG Shenwen, DING Lixin, XIE Datong, et al. Group search optimizer applying opposition-based learning [J]. Computer science, 2012, 39(9): 183-187.
- [14] AL-QUNAIEER F S, TIZHOOSH H R, RAHNAMAYAN S. Opposition based computing-a survey [C]//Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN). Barcelona: IEEE, 2010.
- [15] SURESH K, GHOSH S, KUNDU D, et al. Inertia-adaptive particle swarm optimizer for improved global search [C]//Proceedings of Eighth International Conference on Intelligent Systems Design and Applications. Kaohsiung: IEEE, 2008: 253-258.
- [16] 姜建国, 叶华, 刘慧敏, 等. 融合快速信息交流和局部搜索的粒子群算法 [J]. 哈尔滨工程大学学报, 2015, 36(5): 687-691.
JIANG Jianguo, YE Hua, LIU Huimin, et al. Particle swarm optimization method with combination of rapid information communication and local search [J]. Journal of Harbin engineering university, 2015, 36(5): 687-691.
- [17] GAO Bingkun, REN Xiuju, XU Mingzi. An improved particle swarm algorithm and its application [J]. Procedia engineering, 2011, 15: 2444-2448.
- [18] 许少华, 李新幸. 一种自适应改变惯性权重的粒子群算法 [J]. 科学技术与工程, 2012, 12(9): 2205-2208.
XU Shaohua, LI Xinxing. An adaptive changed inertia weight particle swarm algorithm [J]. Science technology and engineering, 2012, 12(9): 2205-2208.
- [19] 张建伟, 张立伟, 胡颖, 等. 开源机器人操作系统: ROS [M]. 北京: 科学出版社, 2012.
- [20] COUSINS S, GERKEY B, CONLEY K, et al. Sharing Software with ROS [J]. IEEE robotics & automation magazine, 2010, 17(2): 12-14.
- [21] ZAMAN S, SLANY W, STEINBAUER G. ROS-based mapping, localization and autonomous navigation using a Pioneer 3-DX robot and their relevant issues [C]//Proceedings of 2011 Saudi International Electronics, Communications and Photonics Conference (SIEPCP). Riyadh: IEEE, 2011: 1-5.

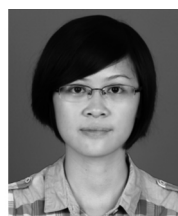
作者简介:



蒲兴成,男,1973年生,副教授,博士,主要研究方向为非线性系统、随机系统和现代智能算法。主持重庆邮电大学校级科研项目3项,参与国际合作项目1项,参与省部级项目6项。发表学术论文30余篇,出版著作1部。



李俊杰,男,1990年生,硕士研究生,主要研究方向为移动机器人自主导航。



吴慧超,女,1990年生,硕士研究生,主要研究方向为智能服务机器人。