

一种改进蚁群算法在捡球机器人多目标路径规划中的应用

曹洁^{1,2}, 耿振节¹¹(兰州理工大学 电气工程与信息工程学院, 兰州 730050)²(甘肃省制造业信息化工程技术研究中心, 兰州 730050)

E-mail: 864470028@qq.com

摘要: 在静态环境下, 针对蚁群算法在进行多目标路径规划时存在搜索时间长以及容易陷入局部最优的特点, 介绍一种可对捡球机器人进行多目标路径规划的改进蚁群算法. 首先采用栅格法将捡球机器人的网球场环境划分建模; 然后自适应调整算法中与信息素相关的系数以及种群规模, 提高算法搜索速度; 并引入交叉操作, 改善算法停滞问题, 增强算法的逃逸能力; 最后实现改进蚁群算法在捡球机器人多目标路径规划中的应用. 仿真实验结果证明, 所提算法可以加快搜索速度, 找到全局最优路径.

关键词: 改进蚁群算法; 捡球机器人; 多目标路径规划; 自适应调整

中图分类号: TP391

文献标识码: A

文章编号: 1000-1220(2015)10-2384-06

An Improved Ant Colony Algorithm and Application in Picking Robot Multi-objective Path Planning

CAO Jie^{1,2}, GENG Zhen-jie¹¹(College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China)²(Technology & Research Center of Gansu Manufacturing Information Engineering, Lanzhou 730050, China)

Abstract: In the static environment, to solve searching for long time and easy to fall into local optimum in ant colony algorithm, an improved ant colony algorithm was proposed for multi-objective path planning of picking robot. First adopts grid method modeling the working tennis court environment of picking robot; Then adaptive adjustment the pheromone related coefficients and population size, improve the search speed of the algorithm, and introduce crossover operation, improve algorithm stagnation problem, enhance algorithm escape ability; Finally achieve improved ant colony algorithm in the application of a multi-objective path planning for picking robot. Simulation results show that the algorithm can speed up the search speed, find the global optimal path.

Key words: improved ant colony algorithm; picking robot; multi-objective path planning; adaptive adjustment

1 引言

移动机器人导航技术是机器人领域研究的核心^[1], 而路径规划是导航技术中的研究热点. 机器人路径规划的任务是根据某些优化准则^[2]找到一条从起点到终点的最优或者接近最优的路径. 目前, 针对移动机器人路径规划的研究, 国内外学者提出了大量算法, 包括人工势场法^[3-4]、禁忌搜索算法^[5]等传统算法, 以及神经网络算法^[6]、遗传算法^[7]、粒子群优化算法^[8]和蚁群算法^[9,10]等智能算法. 由于传统算法搜索效率低, 难以得到理想的路径. 蚁群算法具备其他智能算法没有的并行性和正反馈性的特点, 并且在复杂优化问题的求解中颇具优势, 特别适合机器人的多目标路径规划, 因此目前越来越多的学者开始研究蚁群算法在机器人路径规划领域的应用.

蚁群算法^[11](Ant Colony Optimization Algorithm, ACO)是由Dorigo M等人在1996年提出的模拟自然界真实蚂蚁觅食过程的一种仿生算法, 鲁棒性强, 并且容易和其他方法结合. 在解决旅行商问题(TSP)^[12]、机器人路径规划^[13]和二次

分配(QAP)^[14]等方面有着广阔的应用前景. 由于蚁群算法搜索速度慢且容易陷入局部最优, 文献[9]提出一种根据目标点自适应调整启发函数并借鉴狼群分配原则对信息素进行更新的方法, 在一定程度上可以避免算法陷入局部最优, 但是路径上的信息素变化不够快, 缩短算法搜索时间的能力很有限. 文献[10]动态调整信息素 Q 和信息素挥发系数 ρ , 可以加快算法搜索速度, 但缺乏对算法参数的整体考虑, 容易陷入局部最优.

通过对蚁群算法进行多目标路径规划的深入分析, 发现不同的搜索阶段对算法参数有不同的需求, 利用算法参数因子对算法搜索性能的重要性, 本文提出一种根据迭代次数自适应改变算法参数的改进蚁群算法. 通过模拟蚂蚁的觅食行为, 由数只蚂蚁对多个目标进行协同搜索. 搜索过程中通过自适应调整信息素启发系数 α 、距离启发系数 β 、信息素挥发系数 ρ 、信息素量 Q 以及种群规模 m , 可加快搜索速度, 提高全局搜索能力; 在搜索后期引入交叉操作, 可增强算法的健壮性和逃逸能力. 实验结果表明, 本文所提的改进蚁群算法与基本蚁群算法和遗传算法相比, 在搜索速度和全局搜索能力方面

收稿日期: 2014-07-29 收修改稿日期: 2014-09-17 基金项目: 国家自然科学基金项目(61263031)资助; 国家科技支撑计划课题项目(2012BAF12B19)资助. 作者简介: 曹洁, 女, 1966年生, 教授, 博士生导师, 研究方向为信息融合、智能交通; 耿振节, 男, 1989年生, 硕士研究生, 研究方向为机器人路径规划、嵌入式应用.

都有明显的优势。

2 问题模型和算法描述

2.1 问题模型

2.1.1 问题描述

网球场上散落有一定数量的网球,设定好捡球机器人的起始点和终止点,起始点和终止点各算作一个网球。约束条件:捡球机器人从起始点行走至终止点;无重复经过球场上的所有网球;行走路线的距离尽量短。

2.1.2 模型建立

已知赋权图 $G=(C,L)$,其中 $C=\{c_1, c_2, \dots, c_n\}$ 为目标的顶点集, n 表示问题规模,即网球数量; $L=\{l_{c_i, c_j} | c_i, c_j \in C\}$ 为各顶点相互连接所组成的边集,这里表示集合 C 中网球两两连接所得到的集合。每条边 l_{ij} 对应权值 d_{ij} , d_{ij} 表示边的长度,从而建立起一个 $D=(d_{ij})$ 的矩阵。

此问题中, $d_{ij}=d_{ji}$, 对网球 $C=\{c_1, c_2, \dots, c_n\}$ 的一个访问顺序为 $T=\{t_1, t_2, \dots, t_i, \dots, t_n\}$, 其中 $t_i \in C (i=1, 2, 3, \dots, n)$,

则此问题的数学模型为: $\min(L) = \sum_{i=1}^n d_{t_i, t_{i+1}}$ 。

2.2 算法描述

蚁群算法中两个关键的步骤是目标的概率选择和信息素的更新方式。

2.2.1 概率选择

算法开始运行时,各条路径上的信息素量相同,设 $\tau_{ij}(0)=C_0$ (C_0 为常数)。蚂蚁 $k(k=1, 2, \dots, m)$ 在寻找路径的过程中,根据目标点的能见度和可选择路径上的信息素量做出判断, t 时刻位于网球 i 的蚂蚁 k 移动到网球 j 的概率 $p_{ij}^k(t)$ 为:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{j \in allowed_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}, & j \in allowed_k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

其中, $\tau_{ij}(t)$ 是 t 时刻连接网球 i 与网球 j 的路径上的信息素量; $\eta_{ij}(t)$ 是 t 时刻蚂蚁从网球 i 转移到网球 j 的启发式因子(又叫期望程度或可见度),一般取 $\eta_{ij}=1/d_{ij}$, d_{ij} 为网球 i 和网球 j 之间的距离; $allowed_k$ 为蚂蚁下一步允许选择的所有网球, $allowed_k = \{1, 2, \dots, n\} - \text{tatu}_k$; α 为路径上的信息素量、 β 为启发式因子的重要程度。

2.2.2 信息素的更新

在搜索完所有的网球后,根据式(2)更新搜索到的路径上的信息素。

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

其中,常量 $\rho \in (0, 1)$ 为信息素挥发系数,表示路径上信息素的损耗程度,其大小直接影响算法搜索效率。 $\Delta\tau_{ij}(t)$ 表示本次循环中的路径 (i, j) 上的信息素增量,搜索初期 $\Delta\tau_{ij}(t)=0$ 。

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{蚂蚁 } k \text{ 经过 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (4)$$

其中, $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在本次搜索中对路径 (i, j)

上信息量的影响值。 Q 表示信息素强度,它可以影响到算法的搜索速度; L_k 表示第 k 只蚂蚁在本次搜索中所得路径的长度。

3 改进蚁群算法

基本蚁群算法在路径规划中的应用主要是根据信息素和启发式搜索相结合来完成路径搜索的,在搜索过程中,依靠随机选择策略来构造路径,这样就使得路径搜索速度缓慢。其中信息素的目的在于增大较优路径被选择的概率,但容易使算法陷入局部最优。为解决这些问题,本文对其做出了改进和优化,不仅可以加快搜索速度,还能有效避免算法陷入局部最优。

3.1 种群规模自适应变化

在改进蚁群算法搜索路径的过程中,不同阶段使用不同的种群规模可以加快搜索速度。文献[15]所提出的根据实际情况及时决定种群规模的大小,这样处理虽然可以在前期尽快找到最优解,但是会降低路径的搜索效率。本文的种群规模自适应变化如式(5)所示,其中, M 为初始种群数量, m 为当前种群规模, Nc_{\max} 为最大迭代次数, Nc 为当前已迭代次数。

$$m(t) = \begin{cases} M - 20 \cdot M_{\min} \cdot \frac{Nc}{Nc_{\max}}, & m > M_{\min} \\ M_{\min}, & \text{其他} \end{cases} \quad (5)$$

其中, M_{\min} 为最小种群规模,用来限制种群规模的衰减程度。

3.2 动态调整信息素强度 Q

在用蚁群算法搜索路径的过程中,如果信息素强度 Q 始终为一常数,则 Q 取值的大小对算法性能的影响并不明显;但是如果 Q 在搜索过程中动态变化,则会对算法性能产生一定的影响。本文用动态变化的 $Q(t)$ 代替基本蚁群算法式(4)中信息素强度 Q ,这样可以保证不会因为某条路径上的信息素过多或者过少导致算法停滞,而且可以提高算法的求解精度。

$$Q(t) = \begin{cases} Q_1, & Nc < Nc_{\max}/5 \\ 3Q_1, & Nc_{\max}/5 \leq Nc < 3Nc_{\max}/5 \\ 5Q_1, & 3Nc_{\max}/5 \leq Nc < Nc_{\max} \end{cases} \quad (6)$$

其中, Q_1 表示初始信息素,值取 100。

3.3 自适应权值系数 α 、 β 和信息素残留算子 ρ

权值系数 α 、 β 以及信息素残留算子 ρ 的大小与 ACO 全局搜索能力以及收敛速度相关^[16],对算法的性能影响较大。文献[16]指出, α 的值影响各个目标点上(这里指网球场上的网球)的信息素量对后来蚂蚁引导能力,其值越大,蚂蚁选择信息素量大的路径的可能性就越大; β 的值表示路径距离对蚂蚁的启发程度,其值越大,蚂蚁转移到离它近的目标点的可能性越大; ρ 表示路径上信息素的挥发程度,其值直接影响算法性能。经验表明 ACO 中最优的算法参数组合为 $m=\sqrt{n} \sim n/2$, $\alpha=1 \sim 5$, $\beta=1 \sim 5$, $\rho=0.7$, $Q=100$,基于此,本文中取 α 和 β 自适应函数式为:

$$\alpha = \begin{cases} Nc/M, & Nc \leq Nc_{\max}/2 \text{ 且 } \alpha < 5 \\ 5, & \text{其他} \end{cases} \quad (7)$$

$$\beta = \begin{cases} 9 - Nc/M, & Nc < Nc_{\max} \text{ 且 } \beta > 1 \\ 1, & \text{其他} \end{cases} \quad (8)$$

其中, M 表示初始种群规模; Nc 为当前已迭代次数,即循环次数; Nc_{\max} 是最大迭代次数。

ρ 自适应函数式为:

$$\rho(t+1) = \begin{cases} \delta \cdot \rho(t), & \rho(t) \geq \rho_{\min} \\ \rho_{\min}, & \text{其他} \end{cases} \quad (9)$$

其中, δ 为一常数, 这里取值为 0.97; ρ_{\min} 为 ρ 的最小值, 取值 0.05; 避免因 ρ 过小而降低算法性能。

3.4 算法规则

3.4.1 状态转移规则

蚂蚁采用伪随机比例原则 (pseudo-random-proportional rule) 选择下一个网球, 即位于网球 i 的蚂蚁 k , 以概率 q_0 转移到网球 j , j 是使 $[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$ 达到最大的网球, 蚂蚁的状态转移公式为:

$$j = \begin{cases} \arg \max_{j \in allowed_k} \{ [\tau_{ij}]^\alpha [\eta_{ij}]^\beta \}, & \text{If } q \leq q_0 \\ S, & \text{otherwise} \end{cases} \quad (10)$$

其中, $q_0 \in (0, 1)$ 是事先设定的常数, $q \in (0, 1)$ 是随机数。如果 $q \leq q_0$ 时, 则从所有可行网球中找出使 $[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta$ 最大的网球, 否则, 按式 (1) 以轮盘赌的方式选择下一个网球。

3.4.2 信息素更新规则

首先借鉴最大最小蚂蚁系统 (MMAS) 中的信息素限制, 将所有路径上的信息素的浓度限制在 $[\tau_{\min}, \tau_{\max}]$ 范围内, 由于信息素挥发和更新导致的信息素浓度偏高、偏低的价值都强行限制在 $[\tau_{\min}, \tau_{\max}]$ 之间, 改善算法的停滞现象。

局部信息素更新规则的原则是使已经走过的路径对后面的蚂蚁产生较小的吸引力, 可以有效地避免蚂蚁过早的收敛到同一路径, 对路径 (i, j) 按式 (11) 进行局部信息素更新。

$$\tau_{ij} = (1 - \zeta) \cdot \tau_{ij}(t) + \zeta \cdot \tau_0 \quad (11)$$

其中, ζ 为更新局部信息素的挥发系数, τ_0 为常数, 这里取值为信息素的初值。

全局信息素更新规则是在所有的蚂蚁都完成解的构建后进行的, 只应用于最优的蚂蚁路径上, 全局更新规则参照 (2)。

3.4.3 交叉算子的引入

当算法搜索过程满足交叉条件时, 对得到的路径进行交叉操作。在选取要交叉的路径时, 如果随机选取两条路径交叉处理, 则有可能所选的两条路径均不包含本次搜索中的最优路径或者全局最优路径, 此时交叉操作对算法的影响微乎其微。故为了提高效率, 本文中的交叉是在本次搜索得到的最优路径与随机选择的另外一条路径之间进行, 若两条路径有交叉的网球目标点, 则随机地选择一个相同的目标点进行交叉 (起点和终点除外); 若两条路径没有交叉的目标点, 则不进行交叉处理。如果在交叉后出现了优于本次搜索的最优路径的新路径, 则更新本次搜索中交叉后的最优蚂蚁的路径信息, 否则进行下一次交叉操作。交叉算子的引入, 提高了算法搜索路径的全局能力, 同时也增加了了解的多样性。

4 算法步骤

图 1 所示为改进蚁群算法的流程图, 算法的详细操作方法描述如下。

4.1 实验场景建模

捡球机器人工作在网球场, 环境信息由场内视觉传感器获得。根据图 1 所示的算法流程图, 首先将捡球环境做栅格处理, 建立环境模型, 在算法搜索过程中, 始终将所有路径上的

信息素的浓度限制在 $[\tau_{\min}, \tau_{\max}]$ 范围内。

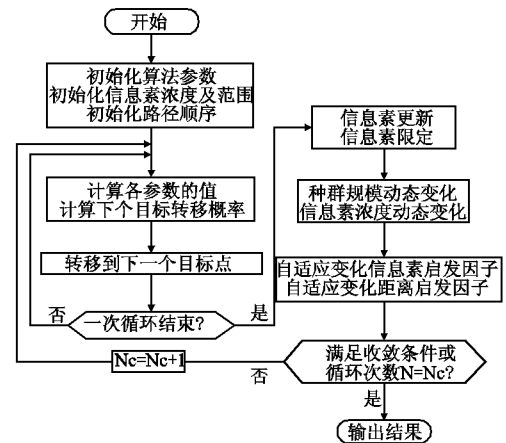


图 1 改进蚁群算法流程图

Fig. 1 Flow chart of improved ant colony algorithm

4.2 初始化算法参数

a) 设置计数器初始值 $Nc = 0$, 最大迭代次数为 Nc_{\max} , 蚂蚁数量初始值为 M , 当前蚂蚁数量为 m , 蚂蚁的起始点为 S_{begin} 。

b) 在起始点放置 M 只蚂蚁, 它们的目标为栅格环境中要经过的每个目标点, 即要捡取的所有网球。

4.3 参数算法迭代

a) 算法进行循环迭代, $Nc = Nc + 1$, 第 k 只蚂蚁按式 (10) 计算的概率选择下一个要行进的节点, 式 (10) 中信息素浓度和启发式信息的权重系数 α, β 分别按式 (7) 和 (8) 计算。

b) 信息素挥发系数 ρ 采用自适应函数 (9) 计算。

c) 计算蚂蚁种群规模的进化情况, 根据式 (5) 调整当前种群规模大小。

d) 信息素强度 Q 按式 (6) 根据时间进行计算。

e) 当蚂蚁前进一步后, 根据式 (11) 进行局部信息素更新。

f) 当所有蚂蚁完成一次周游后, 找出本次迭代中的最优蚂蚁路径、最差蚂蚁路径以及目前为止的全局最优路径, 在得到的本次迭代中的最优蚂蚁路径与随机选择的另外一只蚂蚁路径进行交叉操作。然后对全局最优路径分别按式 (2) 和 (3) 更新对应路径上的信息素, 并对局部最差路径和局部最优路径进行信息素限定。

g) 判断是否满足收敛条件或者已达到最大迭代次数, 满足则转向步骤 4), 否则转向步骤 3)。

4.4 算法终止

算法终止, 输出当前最优路径。

5 仿真实验结果

本文所涉及的捡球机器人是团队自主设计的适用于网球、乒乓球等运动场馆的智能捡球机器人, 由图像采集系统、避障系统、动力系统、捡球计数系统、信息处理系统组成, 属于自主移动捡球机器人。重量约为 4.5kg, 长 90cm, 宽 60cm, 高 50cm。由于机器人运动装置的独特设计, 其可在原地进行任意角度的转向, 不存在转弯半径的问题。下页图 2 所示为智能

捡球机器人的原型样机,图 2 中右下角所示为智能捡球机器人的远程监控界面。

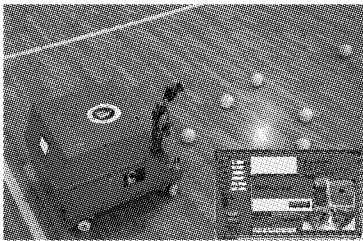


图 2 BALL BOY 机器人
Fig.2 BALL BOY robot

捡球机器人工作环境为网球场,通过场内顶部视觉传感器获取环境信息后,直接对信息进行数据处理,全部环境数据学习完成后,采用栅格法对环境进行划分,栅格大小可根据实际需要和处理能力调整.不考虑机器人在高度方向上的信息,机器人工作的二维空间记为 XY.

表 1 基本参数变化范围
Table 1 Ranges of basic parameters

范围	α	β	ρ	Q	N_c	m
上限	1	1	ρ_{\min}	100	0	M_{\min}
下限	5	5	1	1000	$N_c \max$	M

按国际网球联合会的规定,标准网球场的尺寸为 36.57m * 18.28m,建模后栅格空间为 24 * 15,捡球机器人约占一个栅格。

为了验证本文所提出的改进蚁群算法对于多目标路径规划的有效性和可行性,采用 VC 进行编程仿真,并将仿真结果与基本蚁群算法和遗传算法作比较.算法仿真环境如图 3 所示,基本参数变化范围如表 1 所示.分别对算法进行问题规模为 30 个网球和 40 个网球的实验。

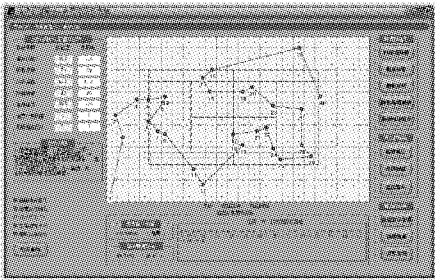


图 3 算法仿真环境
Fig.3 Simulation environment for algorithm

实验约定:在仿真环境中,黑色圆点表示捡球机器人的起始点,灰色圆点表示捡球机器人的终止点,深灰色圆点表示网球场上的散落的网球.在实验中,起始点和终止点也作为目标点之一,且在实际运作中可对起始点和终止点的位置进行调整,以满足实际需求。

5.1 30 个网球的仿真实验

实验重复次数为 20,网球数量 n 为 30.当蚂蚁数量 m 为 20,迭代次数为 100 时路径规划仿真图和收敛曲线如图 4 所示.相同条件下,不同蚂蚁数量和不同迭代次数的详细数据对比结果如下页表 2 所示,表中数据值均为重复 20 次实验的平均值,距离单位为像素点,时间单位为毫秒。

通过路径规划图、收敛曲线图和对比数据表可以看出,在

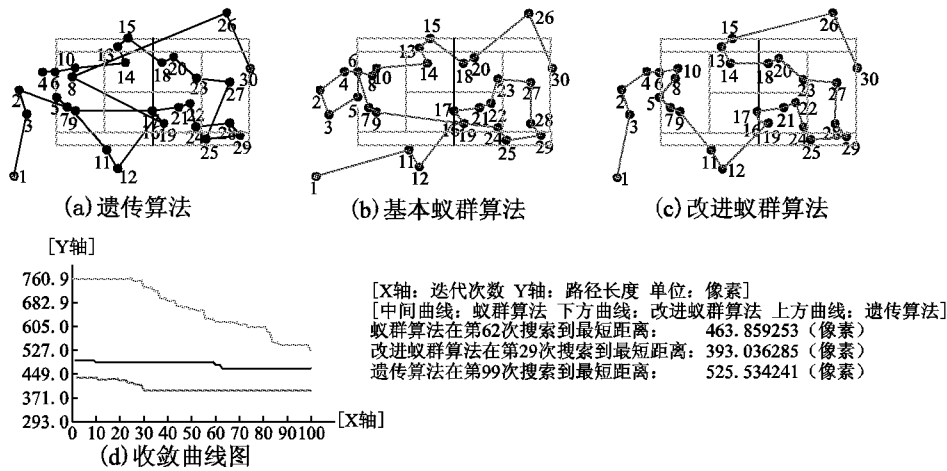


图 4 路径规划和收敛曲线图
Fig.4 Path planning & Convergence graph

网球数量为 30 个时,相同的条件下,不管是最优路径的长度,还是寻优过程所花费的时间,改进蚁群算法与基本蚁群算法相比均有明显的优势.从数据比较可以看出,蚁群算法容易过早停滞,造成所寻找到的路径不是全局最优甚至不是接近全局最优的路径.而遗传算法虽然耗用时间较短,但在同等条件

下同样并未得到全局最优或者接近全局最优的路径.实验结果表明,本文提出的改进蚁群算法可以加快算法搜索速度,减少算法耗用的时间,改善算法的停滞问题.该算法是可行的。

5.2 40 个网球的仿真实验

实验重复次数为 20,网球数量 n 为 40.当蚂蚁数量 m 为

40,迭代次数为 200 时路径规划仿真图和收敛曲线如图 5 所示. 相同条件下,不同蚂蚁数量 and 不同迭代次数的详细数据对比结果表 3 所示,表中数据值均为重复 20 次实验的平均值,

距离单位为像素点,时间单位为毫秒.

通过路径规划图、收敛曲线图 and 对比数据表可以看出,在问题规模进一步扩大,即网球数量为 40 个时,不管是最优路

表 2 同一场景不同条件下算法的数据比较
Table 2 Same scene algorithms' data comparison in different conditions

蚁群数量 迭代次数		10			20			30		
		遗传算法	蚁群算法	改进蚁群算法	遗传算法	蚁群算法	改进蚁群算法	遗传算法	蚁群算法	改进蚁群算法
50	距离	607.983	463.728	405.028	559.628	461.577	402.969	591.671	511.550	400.0671
	时间	54.3645	45.8045	40.7092	60.2228	82.6818	47.5851	56.0526	110.089	47.7972
100	距离	568.337	445.273	398.925	536.647	449.382	396.128	535.7228	502.5580	396.979
	时间	56.4685	73.4728	72.0428	66.0611	142.531	77.7091	67.4134	194.310	79.6737
150	距离	514.510	413.573	396.128	502.830	448.423	394.973	522.4234	505.3859	395.002
	时间	67.6408	98.1401	101.588	68.524	189.005	104.528	66.9283	281.620	110.861
200	距离	475.962	412.775	393.036	464.219	450.120	393.036	443.296	508.643	393.036
	时间	76.7155	127.81	133.941	71.013	253.161	133.525	77.7525	368.945	142.663
300	距离	447.391	417.670	393.036	416.157	446.832	393.036	433.789	510.410	393.036
	时间	77.6234	187.695	180.063	90.074	361.623	197.348	83.702	539.221	209.725

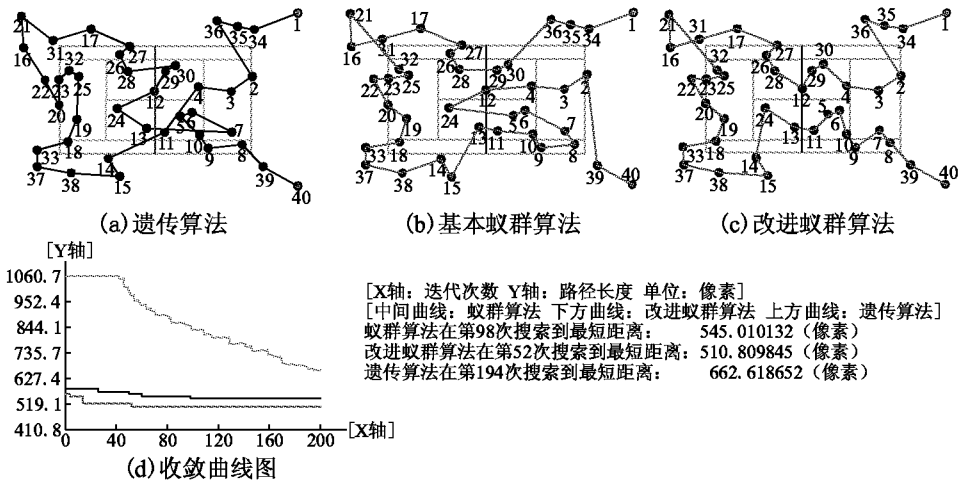


图 5 算法路径规划仿真和对比图
Fig. 5 Path planning & Convergence graph
表 3 同一场景不同条件下算法的数据比较

Table 3 Same scene algorithms' data comparison in different conditions

蚁群数量 迭代次数		10			30			40		
		遗传算法	蚁群算法	改进蚁群算法	遗传算法	蚁群算法	改进蚁群算法	遗传算法	蚁群算法	改进蚁群算法
50	距离	973.416	738.463	532.650	938.378	748.421	524.702	849.034	728.832	523.338
	时间	59.5932	244.498	137.288	58.4078	341.414	133.944	68.1382	460.147	146.253
100	距离	849.386	711.164	525.156	817.754	738.204	518.439	825.778	716.620	520.352
	时间	72.167	343.241	181.152	73.286	494.630	195.497	72.1859	649.250	215.039
150	距离	763.958	690.521	519.284	756.621	683.866	517.334	723.219	592.740	510.809
	时间	78.8525	443.586	232.894	77.7458	682.826	250.925	79.4001	1199.02	280.785
200	距离	710.157	672.377	514.048	670.345	699.782	514.048	655.131	571.526	510.809
	时间	86.2134	553.712	292.571	84.0581	843.585	317.837	88.958	1902.31	345.660
300	距离	653.504	634.088	511.526	699.760	638.773	511.526	622.601	575.237	510.809
	时间	93.0791	657.824	345.522	91.583	1011.67	369.655	95.9153	2725.19	414.575

径的长度,还是寻优过程所花费的时间,改进蚁群算法与基本蚁群算法相比均有具有明显的优势. 和 30 个网球数量的数据相比,可以看到,相同条件下基本蚁群算法更容易陷入局部最

优,并且要耗用更多的时间搜索路径. 而遗传算法的快速搜索并没有弥补其未能搜索到最优路径的缺陷,遗传算法得到的路径的值要比改进蚁群算法的最优路径值大的多,难以满足

实际要求.实验结果表明,本文提出的改进蚁群算法不仅能加快搜索速度,改善停滞问题,更重要的是针对不同的问题规模均有较好的性能和适应性,该算法是可行的.

6 结 论

在分析蚁群算法用于解决机器人多目标路径规划问题时存在诸多不足之处的基础上,提出一种适用于捡球机器人多目标路径规划的改进蚁群算法.通过分析算法搜索路径的运行机理,运用算法中不同的参数值对不同搜索阶段的重要影响,在算法搜索路径的过程中对要改进的参数进行自适应调整,可以解决基本蚁群算法搜索初期的停滞问题,促使算法更有效地找到全局最优路径,提高算法搜索效率.实验结果表明,与基本蚁群算法和遗传算法得到的数据相比,文中所提的改进蚁群算法可以高效快速地得到全局最优路径,优势非常明显,证明了该算法的稳定、高效和实用性.

虽然蚁群算法有诸多优势,但是众多研究文献中表明,从其数学基础上来看,它没有像其他启发式算法那样系统的数学分析方法,而且在理论和实际应用方面仍有许多亟待深入研究的问题.对于本文的研究,机器人的多目标路径规划是基于静态环境展开的,后续将在此基础上,针对复杂的动态环境进行研究,以达到根据实际情况实时对机器人规划出有效的行走路径,进一步提高算法的性能和实用性.

References:

- [1] Deepak B L, Parhi D R, Kundu S. Innate immune based path planner of an autonomous mobile robot [J]. Procedia Engineering, 2012, 38(1): 2663-2671.
- [2] Qu Dao-kui, Du Zhen-jun, Xu Dian-guo, et al. Research on path planning for a mobile robot [J]. Robot, 2008, 30(2): 97-101, 106.
- [3] Oussama Khatib. Realtime obstacle avoidance for manipulators and mobile robots [J]. International Journal of Robotics Research, 1986 (5): 90-98.
- [4] Siciliano B, Sciavicco L. A solution algorithm to the inverse kinematic Problem for redundant manipulators [J]. IEEE Journal of Robotics and Automation, 1998 (4): 403-410.
- [5] Huang L, Lai J F, Hou J, et al. Multicast routing algorithm based on tabu search [J]. Journal of Dalian University of Technology, 2010 (9): 801-805.
- [6] Qian Kui, Song Ai-guo, Zhang Hua-tao. Path planning for mobile robot based on adaptive fuzzy neural network [J]. Journal of Southeast University: Natural Science Edition, 2012, 42(4): 637-642.
- [7] Tuncer A, Yildirim M. Dynamic path planning of mobile robots with improved genetic algorithm [J]. Computers & Electrical Engineering, 2012, 38(6): 1564-1572.
- [8] Zhang Chao, Li Qing, Chen Peng, et al. Improved ant colony optimization based on particle swarm optimization and its application [J]. Journal of University of Science and Technology Beijing, 2013, 35(7): 955-960.
- [9] Liu Chang-an, Yan Xiao-hu, Liu Chun-yang, et al. Dynamic path planning for mobile robot based on improved ant colony optimization algorithm [J]. Acta Electronica Sinica, 2011, 39(5): 1220-1224.
- [10] Lu Yu-fan, Zhang Li. Application of selfadaptive ant colony algorithm in TSP [J]. Microcomputer & its Applications, 2012, 31(17): 78-83.
- [11] Dorigo M, Maniezzo V, Colnari A. Ant system: optimization by a colony of cooperating agent [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1): 29-41.
- [12] Shen Xuan-jing, Liu Yang-yang, Huang Yong-ping, et al. Fast ant colony algorithm for solving traveling salesman problem [J]. Journal of Jilin University: Engineering and Technology Edition, 2013, 43(1): 147-151.
- [13] Wang Chun-miao, Soh Y C, Wang Han. A hierarchical genetic algorithm for path planning in a static environment with obstacles [J]. Electrical and Computer Engineering, 2004, 28(3): 1652-1657.
- [14] Stützgen T, Dorigo M. ACO Algorithms for the quadratic assignment problem new methods in optimization [M]. New York: McGrawHill, 1999: 55-66.
- [15] Wu Z L, Zhao N, Ren G H, et al. Population declining ant colony optimization algorithm and its applications [J]. Expert Systems with Applications, 2009, 36(3): 6276-6281.
- [16] Ye Zhi-wei, Zheng Zhao-bao. Configuration of parameters α , β , ρ in ant algorithm [J]. Geomatics and Information Science of Wuhan University, 2004, 29(7): 597-601.

附中文参考文献:

- [2] 曲道奎, 杜振军, 徐殿国, 等. 移动机器人路径规划方法研究 [J]. 机器人, 2008, 30(2): 97-101, 106.
- [6] 钱 夔, 宋爱国, 章华涛. 基于自适应模糊神经网络的机器人路径规划方法 [J]. 东南大学学报: 自然科学版, 2012, 42(4): 637-642.
- [8] 张 超, 李 擎, 陈 鹏, 等. 一种基于粒子群参数优化的改进蚁群算法及其应用 [J]. 北京科技大学学报, 2013, 35(7): 955-960.
- [9] 柳长安, 鄢小虎, 刘春阳, 等. 基于改进蚁群算法的移动机器人动态路径规划方法 [J]. 电子学报, 2011, 39(5): 1220-1224.
- [10] 卢宇凡, 张 莉. 自适应蚁群算法在求解 TSP 问题中的应用 [J]. 微型机与应用, 2012, 31(17): 78-83.
- [12] 申铨京, 刘阳阳, 黄永平, 等. 求解 TSP 问题的快速蚁群算法 [J]. 吉林大学学报: 工学版, 2013, 43(1): 147-151.
- [16] 叶志伟, 郑肇葆. 蚁群算法中参数 α , β , ρ 设置的研究——以 TSP 问题为例 [J]. 武汉大学学报: 信息科学版, 2004, 29(7): 597-601.