

基于双层蚁群算法和动态环境的 机器人路径规划方法

许凯波^{1,2}, 鲁海燕¹, 黄 洋¹, 胡士娟¹

(1. 江南大学理学院, 江苏无锡 214122; 2. 无锡市玉祁高级中学, 江苏无锡 214183)

摘 要: 针对动态环境未知时变的特点, 提出一种机器人路径规划新方法. 在该方法中, 首先对栅格法建立的环境模型进行凸化处理, 以避免机器人沿规划路径移动时陷入 U 型陷阱, 从而加快路径规划的速度; 其次, 提出双层蚁群算法(DACO), 在每次迭代中先用外层蚁群算法寻找一条路径, 然后以该路径为基础构造一个小环境, 接着在该环境下用内层蚁群算法重新寻优, 若寻得的路径质量更高, 则更新路径并执行本文给出的一种新型信息素二次更新策略; 最后, 针对环境中不同动态障碍物的体积和速度, 提出三种避障策略. 动态环境下, 机器人先由 DACO 算法规划一条静态环境下从起点到终点的全局最优路径, 然后从当前起点开始, 通过自带传感器获取动态环境信息, 并根据需要执行等待、正碰或追尾避障策略, 到达新的起点. 仿真实验表明, 该方法可以在动态环境下实时地为移动机器人规划出一条安全且最短的路径, 是求解移动机器人路径规划问题的一种切实有效的方法.

关键词: 动态障碍物; 移动机器人; 路径规划; 蚁群算法; 栅格法

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0372-2112 (2019)10-2166-11

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2019.10.019

Robot Path Planning Based on Double-Layer Ant Colony Optimization Algorithm and Dynamic Environment

XU Kai-bo^{1,2}, LU Hai-yan¹, HUANG Yang¹, HU Shi-juan¹

(1. School of Science, Jiangnan University, Wuxi, Jiangsu 214122, China;

2. Wuxi Yuqi Senior High School, Wuxi, Jiangsu 214183, China)

Abstract: For the characteristics of being unknown and time-varying of dynamic environment, a new method for mobile robot path planning is proposed in this paper. Firstly, in this method the environment model established by the grid method is convexized in order for the robot to avoid falling into U-shaped traps when moving along the planned path and thus to speed up the path planning. Secondly, a double-layer ant colony optimization (DACO) algorithm is proposed. In each iteration of DACO algorithm, a path is found firstly by the outer ACO algorithm, then on basis of which a small environment is constructed, and then the robot re-plans path by the inner ACO algorithm in the small environment; if the newly obtained path is better, then the global optimal path is updated and a new pheromone secondary update strategy proposed in this paper is executed. At last, three kinds of obstacle avoidance strategies are put forward according to the volume and speed of different dynamic obstacles in the environment. In the dynamic environment, the robot plans a global optimal path from the starting point to the destination with respect to the static environment via the DACO algorithm, then from the current starting point, the robot acquires the dynamic environment's information by its embedded sensor and implements waiting-for-obstacle avoidance strategy, collision avoidance strategy or rear-end collision avoidance strategy when necessary, and then the robot moves to next position as a new starting point. Simulation results show that the proposed can plan a safe and shortest path for mobile robot in real time under dynamic environment and is a practical and effective method for mobile robot path planning.

Key words: dynamic obstacles; mobile robot; path planning; ant colony optimization; grid method

1 引言

路径规划^[1-14]是移动机器人研究领域的重要内容之一,其目的是在有障碍物的环境中为机器人规划一条从起点到终点的最优路径.目前,国内外学者已经在这方面作了诸多的研究,并取得了许多显著的成果,如用遗传算法^[15]和人工势场法成功地求解了该问题.遗传算法虽然全局搜索能力强,但存在产生无效路径的问题;人工势场法在势场合力为零时会出现停滞状态,容易陷入局部最优;蚁群算法^[16]具有正反馈、并行性等优点,已成功用于解决机器人路径规划等问题.

按环境信息是否已知可将移动机器人路径规划问题分为全局信息已知和未知两类;按是否含有动态障碍物可分为静态和动态两类.针对移动机器人在全局信息已知的静态环境下路径规划^[5-9]的研究已经比较成熟,如罗德林等^[6]利用机器人在运动中受到的虚拟人工势场合力及其距目标的距离构造综合启发信息作为蚁群算法的启发信息,然后用蚁群算法规划一条全局最优路径,使机器人在势场合力为零的情况下可以继续进行搜索,而不陷入停滞状态;刘建华等^[7]将人工势场中的力因素转化为局部扩散信息素,提出了一种基于势场蚁群算法的全局路径规划方法,使蚁群倾向于在适应值高的子空间内搜索,该方法具有更快的收敛速度和较强的寻优能力;刘亮^[8]采用将栅格凸化处理的方法解决了机器人在路径规划过程中容易陷入 U 型陷阱的问题,从而提高了路径规划的效率.

实际生活中,机器人的工作环境多是动态的,动态环境具有很多不确定性,机器人很容易与障碍物发生碰撞,因此需要研究更高效实时的算法进行动态环境下的路径规划.移动机器人如何在动态环境下进行路径规划^[10-13]的研究也取得了许多进展,如王哲等^[10]受人工势场法的启发,对蚁群算法进行改进,重新定义蚁群算法中的转移概率,从而加快了算法的寻优速度,然后结合局部路径规划和全局路径规划探讨了一种基于已有最优路径的动态环境避障方法;屈鸿等^[11]通过调整转移概率、限定信息素阈值提出一种改进蚁群算法,并根据动态障碍物不同的运行状态提出相应的避障策略,将两者结合提出了一种机器人路径规划方法;Zhu 等^[12]解决了机器人在工作环境超出自己感知范围、包含未知静态和动态障碍物的路径规划问题.但上述研究对动态障碍物的体积和速度的大小没有作具体说明,或只研究了动态障碍物和移动机器人等大的情况,而实际应用中动态障碍物的大小和速度对避障很关键,不可忽视.针对以上问题,本文提出一种在全局静态环境信息已知但含有未知动态障碍物的路径规划方法——基于改进蚁群算法和动态环境的机器人路径规

划方法.首先引用文献[8]的方法对 U 型陷阱进行凸化处理,然后针对动态环境下移动机器人路径规划问题未知时变的特点提出双层蚁群算法(Double-layer Ant Colony Optimization Algorithm, DACO),用该算法进行静态环境下的路径规划,在机器人沿着最优路径移动时,利用自带的传感器实时监测环境,获取动态障碍物的体积和速度并及时采取相应的避障策略,直到安全抵达终点.

2 环境建模

本文采用栅格法^[13]对机器人的工作环境进行建模.栅格分为两种:自由栅格和障碍栅格.其中白色的栅格是自由栅格,黑色的栅格是障碍栅格.机器人只可以在自由栅格内移动.

此外,为了研究的方便,对移动机器人及其工作环境作出以下几点假设:

(1) 将移动机器人视为一个底面直径等于一个栅格边长的圆柱;

(2) 起点、终点以及环境中静态障碍物的信息已知;

(3) 环境中所有动态障碍物作匀速直线运动,但是不同动态障碍物的体积、移动速度和方向随机产生;

(4) 移动机器人自带传感器,能够感知有限范围内动态障碍物的大小、移动速度和方向;

(5) 移动机器人可在匀速直线运动与停止两种运动状态下切换;

(6) 移动机器人和障碍物以栅格为单位作匀速直线运动.

环境中常常会出现 U 型陷阱,这里采用文献[8]提出的方法,对 U 型陷阱进行凸化处理,如图 1 所示,其中图 1(a)中存在三个 U 型陷阱 U1、U2、U3,图 1(b)是经凸化处理后的环境模型,其中灰色栅格为障碍栅格,这样机器人就无法在这些栅格中移动,从而可以避开陷阱.

3 蚁群算法的改进

蚁群算法(Ant Colony Optimization Algorithm, ACO)^[17,18]是 Dorigo 等根据蚂蚁群体的觅食行为提出的一种智能优化算法,目前已经被成功地应用于解决诸如旅行商问题^[19]、机器人路径规划问题等离散优化问题.但蚁群算法存在收敛速度慢、容易陷入局部最优等缺陷,本文针对这些缺陷以及机器人路径规划问题的特点提出一种改进蚁群算法.改进算法中包含两层蚁群算法:外层蚁群算法进行全局搜索,内层蚁群算法进行局部搜索,故将改进算法命名为双层蚁群算法.双层蚁群算法的改进有以下两点:局部优化策略和信息

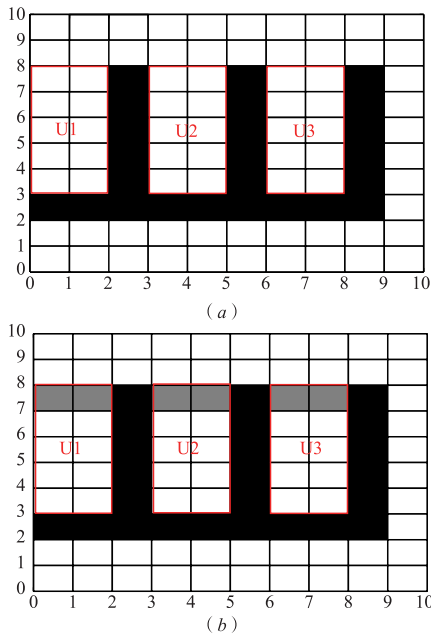


图1 凸化处理

素更新策略.

3.1 局部优化策略

为了提高最优解的精度,将基本蚁群算法全局寻优过程中每次迭代找到的最优路径所在栅格向外扩展一定的距离,从而构造一个小的环境,然后再次调用基本蚁群算法在这个小环境下进行局部寻优.如图2所示,粉色线条是蚁群算法某次迭代找到的最优路径,以该条路径所在的每个栅格为中心向外扩展一个栅格的距离,得到一个如图2中蓝色框所含栅格构成的小环境,然后调用蚁群算法在这个小环境下搜索路径,以寻找精度更高的解.这样充分利用本次迭代找到的最优解来减小搜索空间,虽然增加了算法一次迭代的运行时间,但是,算法可以在较少的迭代次数内找到令人满意的最优解.最终,改进算法可以在更短的时间内找到更精确的解,从而有效提高算法的综合性能.

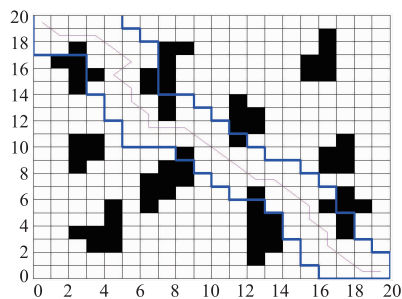


图2 局部优化

3.2 信息素更新策略

在基本蚁群算法中,仅对每次迭代到达终点的蚂蚁找到的路径进行信息素的更新.为了加快算法的收

敛速度,本文提出一种新的信息素二次更新策略:如果局部优化找到更短的路径,则再次更新这条路径上的信息素,从而增加后续蚂蚁探索这条路径的可能性,进一步加快算法的收敛速度.

3.3 双层蚁群算法的步骤

移动机器人用双层蚁群算法进行路径规划的基本步骤如下.

步骤1 机器人对工作环境采用栅格法进行建模,并对U型陷阱进行凸化处理.

步骤2 初始化算法的基本参数,如蚂蚁数量 m 、迭代次数 N_{\max} 、信息素启发因子 α 、期望启发因子 β 、信息素挥发因子 ρ 、信息素强度 Q 等.

步骤3 将所有蚂蚁置于起点,并按式(1)计算转移概率:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [d_{jg}]^\beta}{\sum_{s \in zy_k(t)} [\tau_{is}(t)]^\alpha \cdot [d_{sg}]^\beta}, & s \in zy_k(t) \\ 0, & s \notin zy_k(t) \end{cases} \quad (1)$$

其中 $P_{ij}^k(t)$ 表示 t 时刻蚂蚁 k 从栅格 i 转移到栅格 j 的概率, d_{jg} 表示栅格 j 与终点所在栅格 g 之间的距离, $\tau_{ij}(t)$ 表示 t 时刻栅格 i 与栅格 j 连接路径上的信息素浓度, $zy_k(t)$ 表示 t 时刻蚂蚁 k 可选栅格的集合.

步骤4 将栅格 j 加入禁忌表.

步骤5 重复步骤3到步骤5,直到除迷失蚂蚁外的所有蚂蚁都到达终点,保留每只蚂蚁所走的路径并计算其长度,记录本次迭代的最优路径.

步骤6 按式(2)更新信息素:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k \end{cases}, 0 < \rho < 1 \quad (2)$$

其中, ρ ($0 < \rho < 1$) 表示信息素的挥发程度. $\Delta\tau_{ij}^k$ 表示本次循环第 k 只蚂蚁在栅格 i 和栅格 j 连接路径上释放的信息素, $\Delta\tau_{ij}$ 表示本次循环所有蚂蚁在栅格 i 与栅格 j 连接路径上释放的信息素之和.

步骤7 局部寻优,信息素更新.

步骤8 如果局部寻优找到更短的路径,则对这条路径上的信息素进行二次更新,并更新本次迭代的最优解;否则转到步骤9.

步骤9 若达到最大迭代次数则寻优结束,否则重复步骤3到步骤8.

4 动态环境避碰策略

动态环境中机器人和动态障碍物可能发生以下几种碰撞(本文只考虑作匀速直线运动的障碍物).

情形1 动态障碍物的运行轨迹和机器人的运行

轨迹只有一个交点。

情形 2 动态障碍物的运行轨迹与机器人的运行轨迹有多个交点(即轨迹有重叠),此时又分下面两种情形:

(a) 动态障碍物与机器人的运行方向相反,此时障碍物与机器人会迎面相撞;

(b) 动态障碍物的运行方向和机器人的运行方向相同。

针对上述 3 种情形,本文提出三种相应的动态避障策略。

对于情形 1,采取文献[12]提出的等待策略,即机器人在碰撞点的前一个栅格停止,等待障碍物通过碰撞点

所在栅格后,机器人再移动。但当遇到图 3(a)所示的情形:R 表示机器人,M 表示动态障碍物,实线表示静态环境下的全局最优路径,机器人沿着该路径从起点(*所在栅格)开始移动,障碍物按箭头所指方向移动,两者速度相同,某一时刻机器人通过传感器预测到将会在栅格 P 和障碍物相碰。如果机器人在已规划路径上碰撞点所在栅格的前一个栅格等待障碍物移动一个栅格的时间,即等障碍物移动到栅格 P 后机器人再移动,那么碰撞还是会发生,如图 3(b)(c)所示;如果机器人等障碍物移动两个栅格后再移动,碰撞依然会发生,如图 3(d)(e)所示,只有等障碍物移动 3 个栅格后机器人再移动才能避免发生碰撞,如图 3(f)所示,称此策略为等待避障策略。

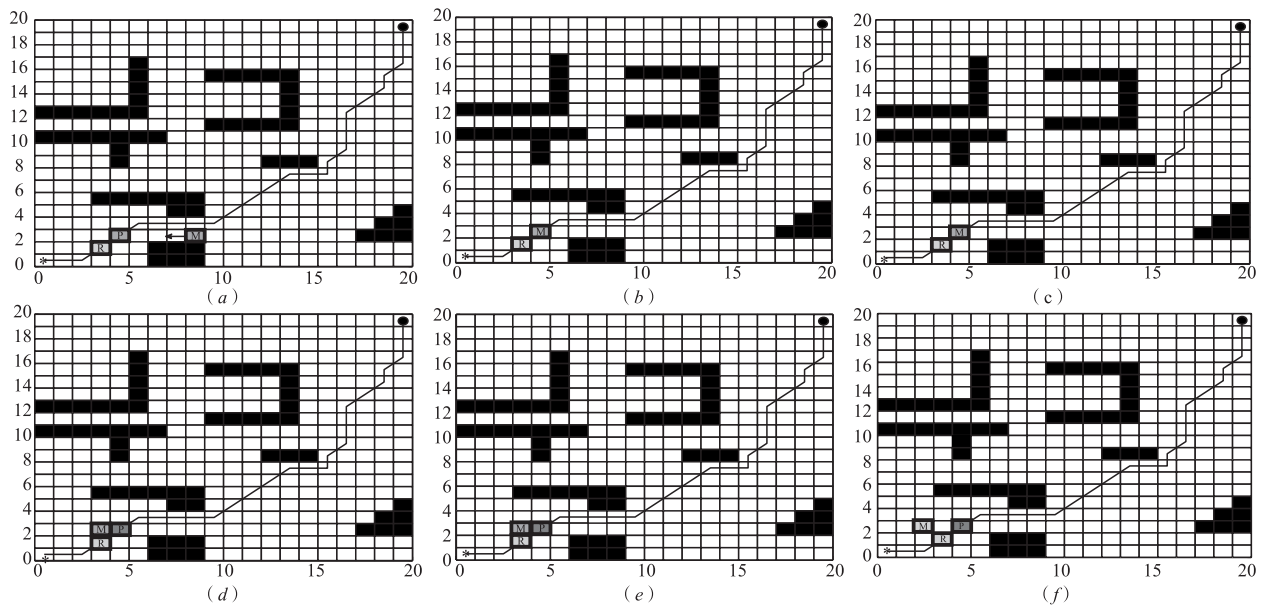


图3 等待避障策略

针对情形 2(a),如果障碍物的大小和机器人一样或小于机器人,已有很多方法,通常将碰撞点视为障碍栅格,然后以碰撞点的前一个栅格为新的起点,以目标点为终点,调用算法规划一条新的路径;但如果障碍物比机器人大,且速度比机器人的速度快。此时的避障策略比较复杂,下面通过 1 个例子来说明。假设障碍物的大小是机器人的 4 倍,速度是机器人的 2 倍,如图 4(a)所示,某一时刻,机器人 R 从图示位置沿着规划的路径移动,同时障碍物 M 从图示位置按箭头所示方向移动,两者移动方向相反,那么经过一定时间,机器人和障碍物会在位置 P 发生碰撞,如果机器人从位置 3 开始规划避障路径,如图 4(b)所示,那么依然会发生碰撞,如图 4(c)所示;如果从位置 2 开始规划避障路径,如图 4(d)所示,也会发生碰撞,如图 4(e)所示;只有按图 4(f)所示,视障碍物所处位置 e 所占栅格为障碍栅格,再次调用 DACO 算法从位置 1 开始规划避障路径才更安全,新规

划的路径如图 4(f)中黑色虚线所示,称此策略为正碰避障策略。

针对情形 2(b),如果机器人的速度大于障碍物的速度,那么碰撞不会发生;如果机器人的速度小于障碍物的速度,会发生追尾而碰撞。如图 5(a)所示,假设某一时刻机器人和障碍物处于图示位置,且障碍物的速度是机器人的 2 倍,那么它们会在同一时刻到达位置 P,但实际应用中,机器人和障碍物在前一个栅格已经发生碰撞,如图 5(b)所示,故不能以前一个栅格为新的起点规划一条到终点的新路径,但以前两个栅格为起点依然会发生碰撞,如图 5(c)所示;此时,应视碰撞点所在栅格及其前两个栅格为障碍栅格,并再次调用 DACO 算法从当前位置为起点规划一条到终点的路径才是最安全的路径,新规划的路径如图 5(d)中黑色实线所示。对移动机器人而言,一条安全无碰的路径更重要,称此策略为追尾避障策略。

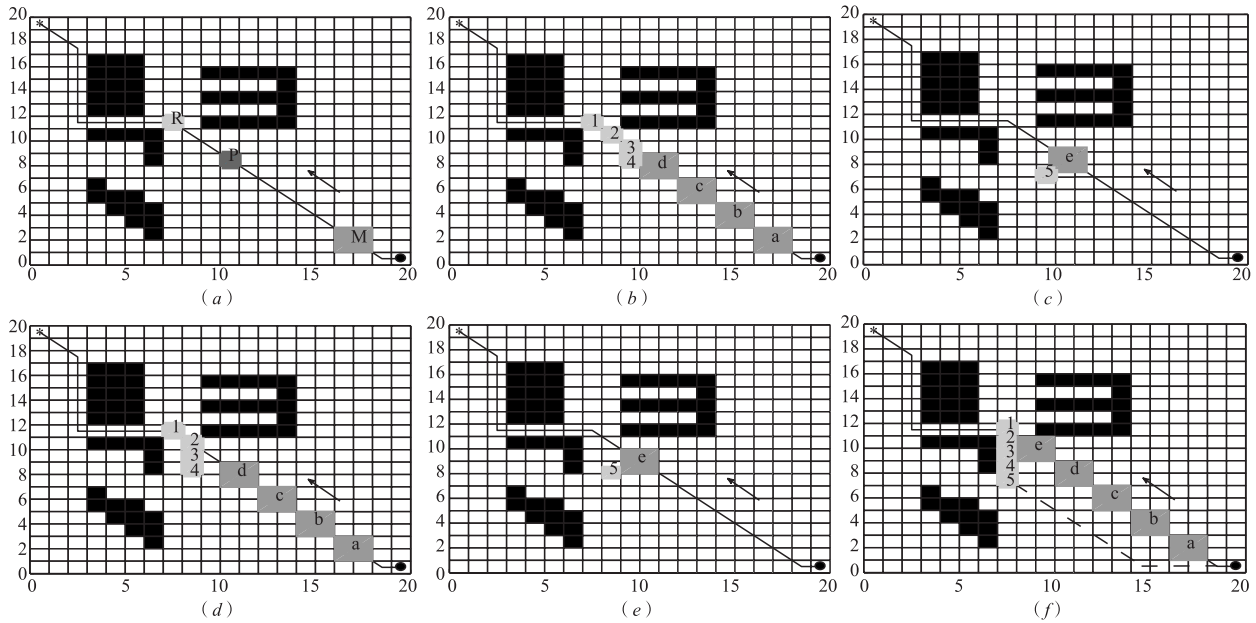


图4 正碰避障策略

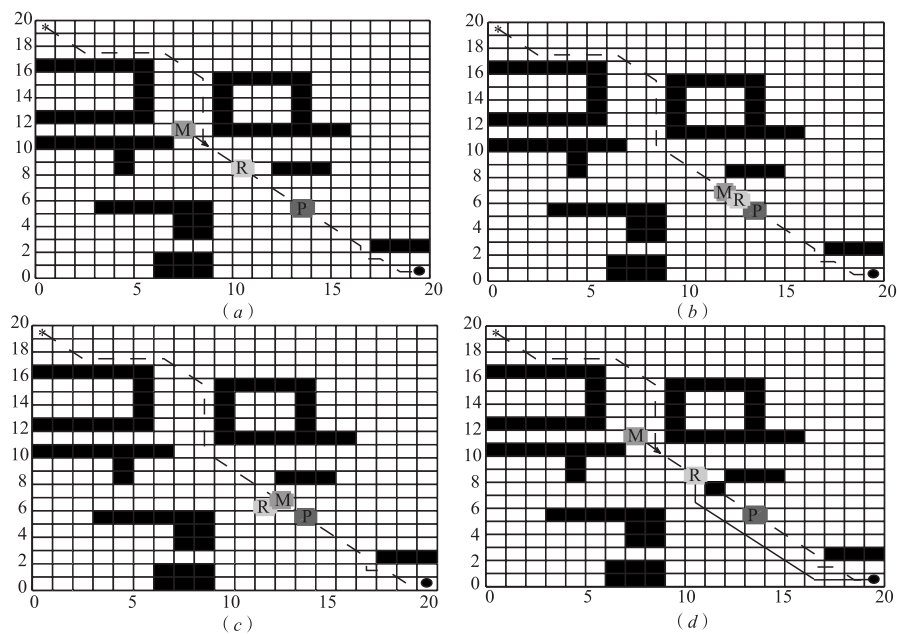


图5 追尾避障策略

5 基于双层蚁群算法和动态环境的路径规划方法

结合第2节环境建模、第3节蚁群算法的改进、第4节动态环境避障策略,本文所提出的基于双层蚁群算法和动态环境的机器人路径规划方法的主要步骤如下。

步骤1 机器人根据已知静态环境信息用栅格法建立环境模型并进行凸化处理。

步骤2 机器人用 DACO 算法规划一条静态环境

下从起点到终点的最优路径 F 。

步骤3 机器人位于当前起点,通过自带传感器获取动态环境的信息,并按第3节所述的动态环境避障策略,根据不同的动态障碍物执行相应的避障策略,到达新的起点;否则沿原路径到达新的起点。

步骤4 机器人判断是否到达终点,若到达,则路径规划结束;否则返回步骤3。

基于 DACO 算法和动态环境的机器人路径规划方法的伪代码(Algorithm)如算法1。

算法 1 DACO 算法伪代码

```

01 采用栅格法建立静态环境模型;
02 对静态环境模型进行凸化处理;
03 采用 DACO 算法为机器人规划一条从起点到终点的全局静态最优
    路径;
04 机器人位于起点;
05 While 机器人没有到达终点
    机器人检测环境信息;
If 情形一
    执行等待避障策略;
    并沿着最优路径向前移动一格;
Elseif 情形二
    执行正碰避障策略;
    更新全局最优路径并沿着最优路径向前移动一格;
Elseif 情形三
    执行追尾避障策略;
    更新全局最优路径并沿着最优路径向前移动一格;
Else
    沿着当前最优路径向前移动一格;
End
06 结束.

```

6 仿真与分析

6.1 凸化处理

如图 6(a) 所示的栅格环境中三个 U 型陷阱 U1、U2、U3, 在没进行凸化处理前, DACO 算法规划的路径如图 6(a) 所示, 跳过了陷阱 U1, 但掉进了陷阱 U2、U3. 如果机器人按图 6(a) 所示路径行走, 显然该路径不是最短路径. 但若将环境中的 U 型陷阱 U1、U2、U3 进行封口处理(见图 6(b)), 视这些灰色栅格为障碍栅格, 然后再用 DACO 算法规划路径, 则最终找到的路径如图 6(b) 所示, 可见该路径成功避开了 U 型陷阱. 两者对比, 显然机器人沿着图 6(b) 所示路径行走路程更短.

6.2 双层蚁群算法与其它蚁群算法的比较

为了验证双层蚁群算法的性能, 本文作了很多仿真实验, 实验结果也验证了本文所提改进算法具有更快的收敛速度和更强的全局寻优能力. 实验在 CPU 为 i5-430M、RAM 为 2G 的计算机上进行, 其中改进算法使用 Matlab 编程, 通过实验验证, 参数设置如表 1 所示时算法性能最优.

表 1 仿真实验参数

m	N_{\max}	α	β	ρ	Q
10	50	1	40	0.8	1

6.2.1 双层蚁群算法与基本蚁群算法的比较

表 2 给出了双层蚁群算法和基本蚁群算法在图 7 所示环境下进行 10 次路径规划的结果, 其中 l_1 是双层

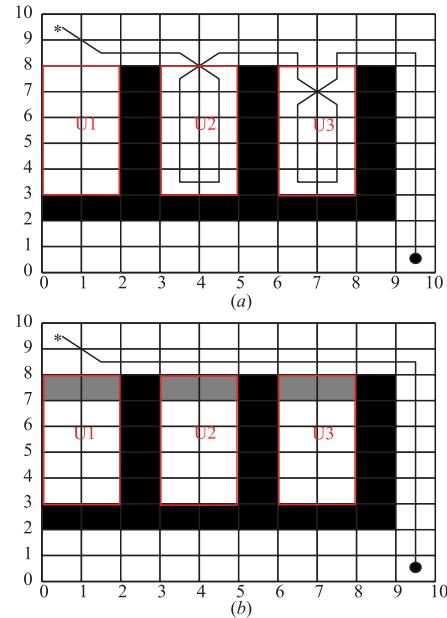


图 6 凸化处理前后路径对比

蚁群算法每次迭代找到的最优路径的长度, l_2 是基本蚁群算法每次迭代找到的最优路径的长度. 从表 1 中可以看出双层蚁群算法有 3 次找到了最优解 32.1421, 而基本蚁群算法找到的最优解为 33.5563, 不是该环境下的最优解. 图 7(a) 是基本蚁群算法找到的最优路径, 图 7(b) 是双层蚁群算法找到的最优路径. 表 2 还统计了两种算法迭代 50 次的时间, 虽然 DACO 算法的运行时间比基本 ACO 算法长, 但 DACO 算法收敛更快, DACO 算法在第 4 代就收敛, 而 ACO 算法在第 10 代才收敛, 且没有收敛到最优值, 综合时间和解的质量来看, DACO 算法的性能更优.

表 2 环境 1 下 DACO 算法和基本 ACO 算法最短路径长度对比

序号	l_1	时间/s	l_2	时间/s
1	33.5563	3.94	34.1421	2.38
2	32.1421	3.11	34.7279	2.38
3	32.1421	3.87	34.9706	2.38
4	32.7279	3.79	34.9706	2.19
5	33.5563	3.98	34.9706	2.19
6	32.1421	3.01	33.5563	2.20
7	33.5563	3.98	34.7279	2.50
8	33.5563	3.08	33.5563	2.23
9	33.5563	3.02	33.5563	2.42
10	33.5563	3.08	34.9706	2.27
平均值	33.0492	3.48	34.4149	2.3

图 8 是两种算法在图 7 所示环境下的收敛图, 从图中可以看出双层蚁群算法在第 4 代收敛到图 7 所示环

境的最优解 32.1421,而基本蚁群算法找到的最优解为 34.7279,比该环境的最优解 32.1421 多 2.5858,在第 10 代才收敛.由此表明,和基本 ACO 算法相比,双层蚁群算法的收敛速度更快.

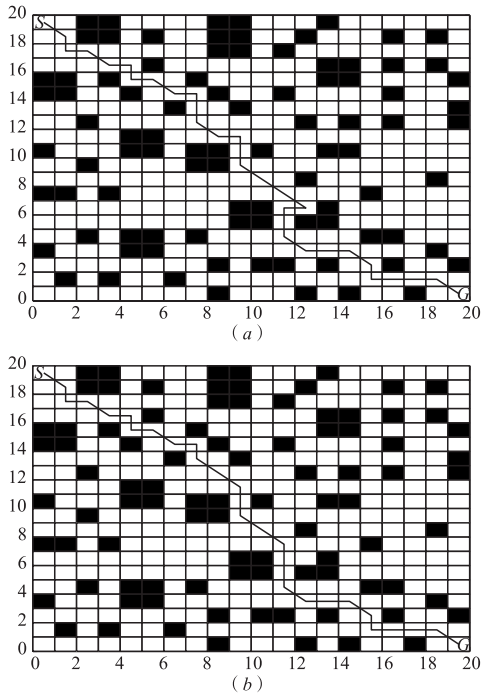


图7 环境1下DACO算法和基本ACO算法最优路径对比

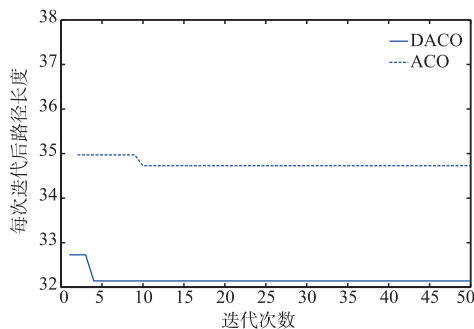


图8 环境1下DACO算法和基本ACO算法收敛速度对比

表3给出了基本 ACO 算法和 DACO 算法在图9所示环境下 10 次路径规划的结果.相比图7,图9所示环境空间更大,复杂度也更高.表中 l_3 表示 DACO 算法找到的最短路径的长度, l_4 表示基本 ACO 算法找到的最短路径的长度; x_3 表示 DACO 算法第一次找到最短路径时的迭代次数, x_4 表示基本 ACO 算法第一次找到最短路径时的迭代次数.从表中可以看出,DACO 算法在 10 次实验中都找到了图9所示环境下的最优路径,而基本 ACO 算法没有找到最优路径,其找到的最优路径长度为 45.6985,比该环境下的最优路径还长 0.5858,表明 DACO 算法的全局寻优能力更强;且第一次找到最短路径时的平均迭代次数也更少,DACO 算法为 4.9

次,基本 ACO 算法为 6 次,再次表明相比基本 ACO 算法,DACO 算法的寻优速度更快.图9(a)为基本 ACO 算法找到的最优路径,图9(b)为 DACO 算法找到的最优路径.

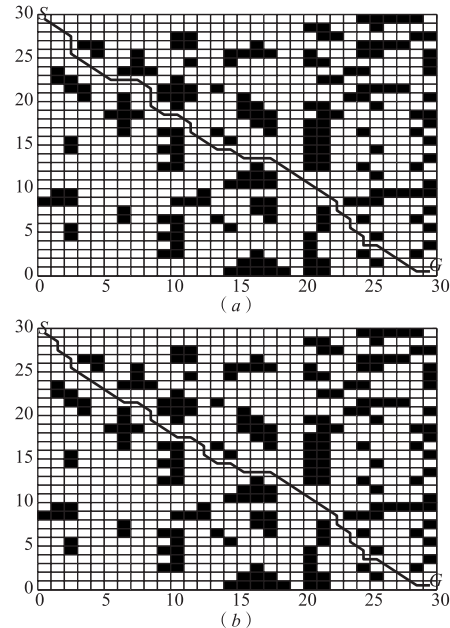


图9 环境2下DACO算法和基本ACO算法最优路径对比

表3 环境2下 DACO 算法和基本 ACO 算法性能比较

序号	l_3	x_3	l_4	x_4
1	45.1127	8	46.5269	6
2	45.1127	1	47.1127	6
3	45.1127	6	45.6985	7
4	45.1127	1	45.6985	7
5	45.1127	4	45.9411	5
6	45.1127	3	46.5269	6
7	45.1127	6	46.5269	6
8	45.1127	2	45.6985	6
9	45.1127	1	46.5269	5
10	45.1127	5	45.6985	6
平均值	45.1127	3.7	46.1955	6

6.2.2 双层蚁群算法与其他改进蚁群算法的比较

为了进一步验证双层蚁群算法的有效性,这里和文献[6,7]中提出的改进蚁群算法进行了比较,图10图11是文献[6]和文献[7]中机器人的工作环境模型.

表4给出了 DACO 算法在环境3和环境4下独立重复 10 次实验的结果,其中 l_5 、 l_6 分别表示 DACO 算法在环境3下每次实验找到的最优路径的长度, x_5 、 x_6 分别表示在每次实验中 DACO 算法第一次找到最优解时的迭代次数.从表4可以得出 DACO 算法在较少的迭代

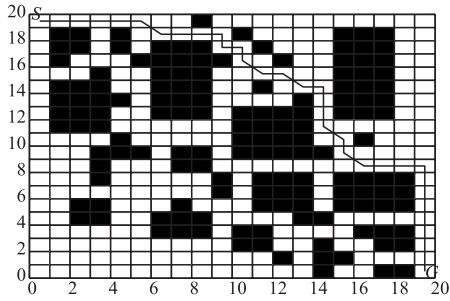


图10 环境3

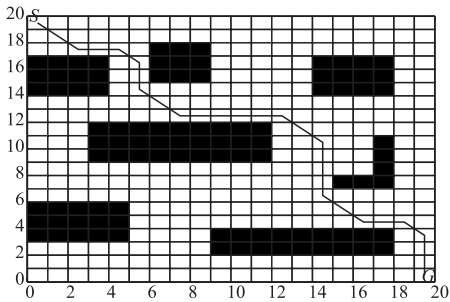


图11 环境4

次数内找到了两种环境的最优路径,说明了 DACO 算法具有较快的寻优速度. 表 5 给出了 DACO 算法和文献 [6,7] 所提算法的对比结果, l_7 、 l_8 分别表示三种算法在环境 3 和环境 4 下找到的最优路径的长度, x_7 、 x_8 分别表示三种算法在环境 3 和环境 4 下第一次找到最优解时的迭代次数. 从表 5 可知, 三种算法都可以找到所给环境的最短路径, 如图 10 图 11 所示, 但是双层蚁群算法可以在更少的迭代次数内找到最优解, 证明了双层蚁群算法的有效性.

表 4 环境 3、环境 4 下 DACO 算法性能测试

序号	环境 3		环境 4	
	l_5	x_5	l_6	x_6
1	32.1421	6	35.0711	1
2	32.1421	1	35.0711	1
3	32.1421	2	35.0711	1
4	32.1421	1	35.0711	2
5	32.1421	2	35.0711	1
6	32.1421	2	35.0711	2
7	32.1421	3	35.0711	1
8	32.1421	2	35.0711	1
9	32.1421	4	35.0711	1
10	32.1421	6	35.0711	1
平均值	32.1421	2.9	35.0711	1.2

表 5 DACO 算法和文献 [6,7] 所提算法比较

算法	环境 3		环境 4	
	l_7	x_7	l_8	x_8
文献 [6]	32.1421	11	35.0711	12
文献 [7]	---	---	35.0711	7.7
DACO	32.1421	2.9	35.0711	1.2

6.2.3 双层蚁群算法与遗传算法、粒子群算法的比较

本文又与遗传算法、粒子群算法进行了比较, 表 6 记录了 DACO 算法与 PSO 算法在环境 5 下的实验结果, 表 7 记录了 DACO 算法和 GA 算法在环境 6 下的实验结果. 从表 6 可以得出: 和 PSO 算法相比, DACO 算法寻优时间更短, 找到的最优路径也更短, 每次都可以找到最优路径, 成功率更高. 从表 7 可以得出: 和 GA 算法、改进 GA 算法相比, DACO 算法的寻优时间更短, 效率更高, 且稳定性高, 进一步说明了改进算法的有效性.

表 6 DACO 算法和 PSO 算法比较

算法	最优路径长度	平均值	时间/ms
PSO	33.8995	37.5710	320
DACO	32.1421	32.1421	209

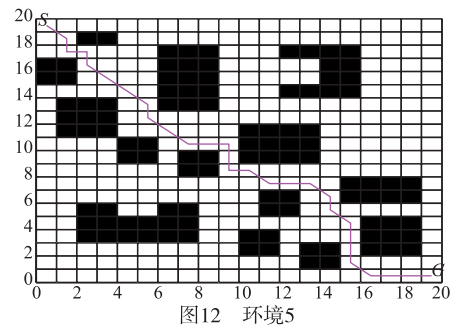


图12 环境5

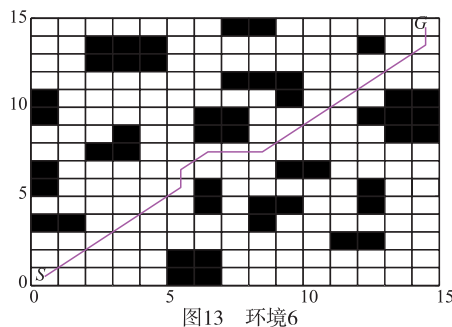


图13 环境6

表 7 DACO 算法和 GA 算法比较

算法	最优路径长度	时间/ms	最优路径比例/%
GA	25.0711	345	62
改进 GA	20.9706	217	95
DACO	20.9706	205	100

6.3 动态环境仿真实验

为了进一步验证本文所提方法的可行性,下面通过一个仿真实例进行说明. 在该实例中,机器人(用蓝色栅格代表)在一个 40×40 的栅格环境中移动,起点坐标为 $(1,40)$, 终点坐标为 $(40,1)$, 如图 14 所示.

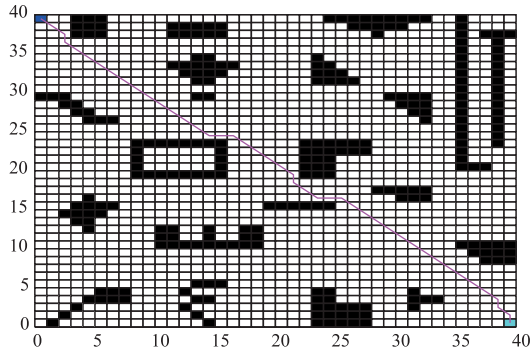


图14 机器人静态环境和DACO算法规划的全局最优路径

首先机器人用双层蚁群算法规划一条静态环境下从起点到终点全局最优路径,如图 14 中的粉色线条所示,然后机器人沿着该路径移动,并且每移动一个栅格检测一次环境信息,以根据环境的变化执行相应的避障策略.

在 t_1 时刻,机器人移动到如图 15(a) 所示位置,通过传感器发现一动态障碍物 M1 (M1 占据 9 个栅格) 以每步 3 个栅格的速度向自己移动过来,并会在红色栅格发生碰撞,此时执行正碰避障策略:机器人以自己当前所在栅格为新的起点,重新规划到终点的路径,新规划的路径如图 15(a) 粉色虚线所示.

在 t_2 时刻,机器人移动到如图 15(b) 所示位置,检测到另一个动态障碍物 M2 (M2 占据 4 个栅格) 以每步 1 个栅格的速度移动,并且预测到会和自己在红色栅格发生侧面碰撞,此时执行等待避障策略:机器人停止移动,等 M2 移动到如图 15(c) 所示位置时再移动.

在 t_3 时刻,机器人检测到另一动态障碍物 M3 (M3 占据 1 个栅格) 以每步 2 个栅格的速度从身后向自己移动,且速度是自己的 2 倍,会在红色栅格发生追尾 (见图 15(d)), 此时执行追尾避障策略,以当前所在栅格为起点,重新规划一条到终点的路径,如图 15(d) 紫色虚线所示.

图 16 为在仿真环境下本文所提出的路径规划方法指引机器人获得的全局无碰最优路径.

7 结束语

本文对动态环境下移动机器人的路径规划问题进行了研究. 首先对栅格法建立的环境模型进行凸化处理,以避免机器人沿规划路径移动时掉入 U 型陷阱,从而加快算法的收敛速度. 此外,提出一种改进蚁群算

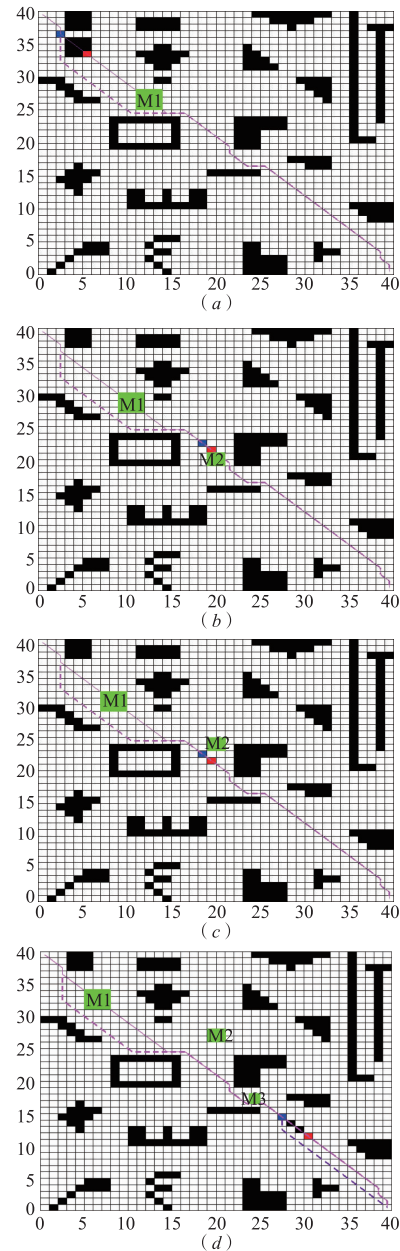


图15 动态环境下机器人路径规划过程

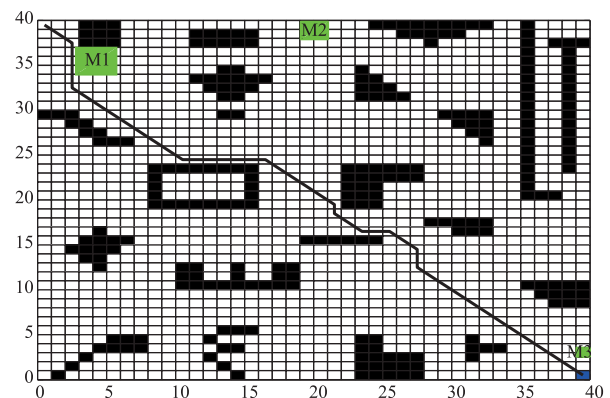


图16 动态环境下机器人最终的全局无碰路径

法——双层蚁群算法,将基本蚁群算法每次迭代找到的最优路径向外扩展一定的距离,然后再次调用基本蚁群算法在这个子空间内寻优,以提高算法的寻优精度;如果找到更优的路径,则再次更新该条路径上的信息素.最后机器人沿着由改进蚁群算法在已知静态环境下规划的全局最优路径移动的过程中,实时感知环境的变化,根据动态障碍物体积、速度的不同自适应避障,最终获取一条从起点到终点的无碰最优路径.通过仿真实验验证了本文所提方法确实可以快速地为移动机器人在动态环境下规划出一条距离最短且安全性更高的路径,是动态环境下机器人路径规划问题的一种有效方法.本文只考虑了二维环境下机器人的路径规划问题,今后将尝试研究把本文的方法推广和应用到三维环境下的机器人路径规划问题.

参考文献

- [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 Daqi, YAN Minzhong. Survey on technology of mobile robot path planning[J]. Control and Decision, 2010, 25(7):961-967. (in Chinese)
- [2] 李宝磊,吕丹桔,张钦虎,等. 基于多元优化算法的路径规划[J]. 电子学报,2016,44(9):2242-2247.
LI Bao-lei, LÜ Dan-ju, ZHANG Qin-hu, et al. A path planner based on multivariate optimization algorithm[J]. Acta Electronica Sinica, 2016, 44(9):2242-2247. (in Chinese)
- [3] 强宁,高洁,康凤举. 基于 PSO 和三次样条插值的多机器人全局路径规划[J]. 系统仿真学报,2017,29(7):1397-1404.
QIANG Ning, GAO Jie, Kang Fengju. Multi-robots global path planning based on PSO algorithm and cubic spline[J]. Journal of System Simulation, 2017, 29(7):1397-1404. (in Chinese)
- [4] 杜鹏桢,唐振民,陆建峰,等. 不确定环境下基于改进萤火虫算法的地面自动驾驶全局路径规划方法[J]. 电子学报,2014,42(3):616-624.
DU Peng-zhen, TANG Zhen-min, LU Jian-feng, et al. Global path planning for ALV based on improved glowworm swarm optimization under uncertain environment[J]. Acta Electronica Sinica, 2014, 42(3):616-624. (in Chinese)
- [5] 康冰,王曦辉,刘富. 基于改进蚁群算法的搜索机器人路径规划[J]. 吉林大学学报(工学版),2014,44(4):1062-1068.
KANG Bing, WANG Xihui, LIU Fu. Path planning of searching robot based on improved ant colony algorithm[J]. Journal of Jilin University(Engineering and Technology Edition), 2014, 44(4):1062-1068. (in Chinese)
- [6] 罗德林,吴顺祥. 基于势场蚁群算法的机器人路径规划[J]. 系统工程与电子技术,2010,36(6):1277-1280.
LUO Delin, WU Shunxiang. Ant colony optimization with potential field heuristic for robot path planning[J]. Systems Engineering and Electronics, 2010, 36(6):1277-1280. (in Chinese)
- [7] 刘建华,杨建国,刘华平,等. 基于势场蚁群算法的移动机器人全局路径规划方法[J]. 农业机械学报,2015,46(9):18-27.
LIU Jianhua, YANG Jianguo, LIU Huapin, et al. Robot global path planning based on ant colony optimization with artificial potential field[J]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(9):18-27. (in Chinese)
- [8] 刘亮. 基于改进蚁群算法的移动机器人路径规划[D]. 武汉:武汉科技大学信息学院,2007.
LIU Liang. Mobile Robot's Path Planning Based on Improved Ant Colony Algorithm[D]. Wu Han: Wuhan University of Science and Technology, 2007. (in Chinese)
- [9] LIU Jianhua, YANG Jianguo, LIU Huaping, et al. An improved ant colony algorithm for robot path planning[J]. Soft Computing, 2016, 21(19):1-11.
- [10] 王哲,孙树栋,曹飞祥. 动态环境下移动机器人路径规划的改进蚁群算法[J]. 机械科学与技术,2013,32(1):42-46.
WANG Zhe, SUN Shudong, CAO Feixiang. An improved ant colony algorithm for mobile robot path planning under dynamic environment[J]. Mechanical Science and Technology for Aerospace Engineering, 2013, 32(1):42-46. (in Chinese)
- [11] 屈鸿,黄利伟,柯星. 动态环境下基于改进蚁群算法的机器人路径规划研究[J]. 电子科技大学学报,2015,44(2):260-265.
QU Hong, HUANG Liwei, KE Xing. Research of improved ant colony based on robot path planning under dynamic environment[J]. Journal of University of Electronic Science and Technology of China, 2015, 44(2):260-265. (in Chinese)
- [12] ZHU Qingbao, HU Jun, CAI Wenbin. A new robot navigation algorithm for dynamic unknown environments based on dynamic path re-computation and an improved scout ant algorithm[J]. Applied Soft Computing, 2011, 11:4667-4676.
- [13] 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.
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. (in Chinese)
- [14] BORENSTEIN J, KOREN Y. Histogramic in-motion mapping for mobile robot obstacle avoidance [J]. IEEE Transactions on Robotics and Automation, 1991, 7(4): 535 - 539.
- [15] 邢焕来, 潘炜, 邹喜华. 一种解决组合优化问题的改进型量子遗传算法 [J]. 电子学报, 2007, 35(10): 1999 - 2007.
- XING Huan-lai, PAN Wei, ZOU Xi-hua. A novel improved quantum genetic algorithm for combinatorial optimization problem [J]. Acta Electronica Sinica, 2007, 35(10): 1999 - 2007. (in Chinese)
- [16] 马卫, 朱庆保. 求解函数优化问题的快速连续蚁群算法 [J]. 电子学报, 2008, 36(11): 2120 - 2124.
- MA Wei, ZHU Qing-bao. Fast continuous ant colony optimization algorithm for solving function optimization problem [J]. Acta Electronica Sinica, 2008, 36(11): 2120 - 2124. (in Chinese)
- [17] DORIGO M, GAMBARDELLA M. Ant colonies for the traveling salesman problem [J]. BioSystems, 1997, 43: 73 - 81.
- [18] DORIGO M, DI CARO G. Ant colony optimization: A new meta-heuristic [A]. Proceedings of the 1999 Congress on Evolutionary Computation [C]. USA: IEEE, 1999. DOI: 10.1109/CEC.1999.782657.
- [19] 许凯波, 鲁海燕, 程毕芸, 等. 求解 TSP 的改进信息素二次更新与局部优化蚁群算法 [J]. 计算机应用, 2017, 37(6): 1686 - 1691.
- XU Kaibo, LU Hanyan, CHENG Biyun, et al. Ant colony optimization algorithm based on improved pheromones double updating and local optimization for solving TSP [J]. Journal of Computer Application, 2017, 37(6): 1686 - 1691. (in Chinese)

作者简介



许凯波 男. 1992 年 6 月出生, 山西晋城人. 硕士, 2018 年毕业于江南大学理学院应用数学专业, 主要研究方向为最优化与控制.
E-mail: 2587073839@qq.com



鲁海燕 女. 1970 年出生, 山东淄博人. 副教授, 硕士生导师, 分别于 1996 年和 2007 年在浙江大学获得硕士学位和博士学位, 现为江南大学理学院信息与计算科学系党支部书记, 主要从事组合优化; 网络优化; 计算智能和机器学习及其在组合优化、计算流体力学、生物信息学中的应用等方面的研究.
E-mail: luhaiyan@jiangnan.edu.cn