

基于评估和分工合作并行蚁群机器人路径规划

吕凌, 曾碧

(广东工业大学, 广东 广州 510006)

摘要:针对复杂环境中移动机器人的导航中存在的问题,提出了一种适用于机器人路径规划的并行蚁群分工合作算法。该方法由控制中心和独立的运算单元组成,每个运算单元中使用分工合作的蚁群进行计算从而从局部和全局两个方面优化蚁群的路径搜索,并将计算发送给处于控制中心的计算机,控制中心则负责处理每个运算单元发送的阶段性的路径搜索结果并利用评估机制对每个计算机得出的结果做最后的决策。从仿真结果可以看出该算法是有效且可行的。

关键词:移动机器人;蚁群算法;评估机制;并行;分工

中图分类号:TP24

文献标识码:A

文章编号:1673-629X(2011)09-0010-04

Path Planning for Robot Introduction Parallel Ant Colony Algorithm Based on Division of Labor and Assessment

LÜ Ling, ZENG Bi

(Guangdong University of Technology, Guangzhou 510006, China)

Abstract: In order to efficiently solve the problem of mobile robot's path planning in complex dynamic environment, proposed a new method which implemented parallel ant colony optimization algorithm based on assessment and division-cooperation of labor. This method consists of control center and independent process unit. Each process unit uses division-cooperation of labor ant colony to optimize the ant search in local and global aspect, and then sends the result to the control center. Control center was responsible for coordinating the result which got from each independent process unit and used the assessment mechanism to make final decision. Simulation result shows that this algorithm is feasible and effective.

Key words: mobile robot; ant colony algorithm; assessment mechanism; parallel; division of labor

0 引言

机器人路径规划就是在复杂的环境空间中,寻找一条由机器人当前位置到环境中某一点的不碰撞并且最短的路径。由于这类问题属于 NP-Hard 问题,因此,如何求得对这个问题的更优解成为了国内外的研究热点^[1,2]。针对这个问题已经开展了广泛的研究,提出了许多著名的算法,如人工势场法^[3]、概率路图法(PRM)^[4]、遗传算法^[5]等。人工势场法的优点是可应用在动态环境问题上,但缺点是容易陷入局部极值。概率路图法适合于多自由度的机器人规划,很难扩展到动态环境中。而遗传算法存在收敛速度慢等的问题。

蚁群优化算法(ACO)是受自然界蚂蚁行为的启发而提出来的。蚂蚁通过在其行进过程中留下一一种称

之为信息素的物质,后面的蚂蚁可以感知到这种物质的强弱,从而决定自己的路径。因此,路径越短,路径上累积的信息素越多,蚂蚁就有越大的概率选择这条路径。这就形成了一个非常有效的正反馈机制。由上面的介绍可以看到,一只蚂蚁的智能不高,但是蚁群却表现出了很高的智能。基本的蚁群算法就是基于这种原理提出来的。

使用蚁群算法解决机器人在复杂环境中的寻径问题可以得到不碰撞的较优路径,但是存在着计算量大、运算时间长、收敛较慢的缺点,不利于提升机器人导航的实时性^[6,7]。文中提出一种存在动态和静态障碍物的环境中的机器人导航方法。这种方法利用了多个独立的处理单元并行计算并相互协作加快机器人路径搜索,并且利用评估机制有效地避开了障碍物并获得了较优的路径。

收稿日期:2011-02-15;修回日期:2011-05-21

基金项目:广东省自然科学基金资助项目(05001801)

作者简介:吕凌(1984-),男,广东韶关人,硕士,研究方向为智能规划与决策;曾碧,教授,硕士研究生导师,研究方向为智能机器人技术、智能规划与决策。

1 算法模型

在机器人的行进过程中,由于传感器的探测范围的限制,只能探测到以机器人中心,半径以机器人传

传感器探测距离为长度的圆形区域^[8]。假设机器人用先验性知识获得了最终目标点的位置,因此,创建最终目标点的位置与机器人的连线相交且垂直的半圆,并在其中使用蚁群进行路径搜索。

在蚁群路径搜索的过程中,存在着路径长度和与障碍物碰撞概率的一个考虑^[9],比如,当机器人在路径搜索过程中障碍物处于机器人及机器人所要达到的路径点之间的这种情况,由于传感器误差、噪声等因素的影响,机器人获取的环境信息与真实环境存在着一定的差别,因此,障碍物周围存在着一个不确定区域,在这个不确定区域内的路径规划就是在该区域内权衡路径长度和碰撞风险的过程^[10,11]。文中通过评估机制对这种不确定区域进行评估,最终决定机器人的行进方向。

算法框架如图1所示。

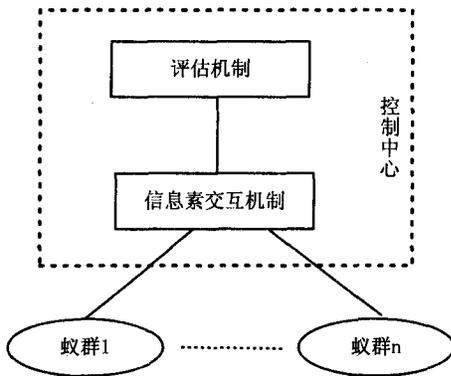


图1 算法框架

其中,蚁群分别运行在不同的处理单元上,在算法迭代到某一固定时间间隔后,各个蚁群分别将迄今为止找到的最优解发送到处于控制中心的计算机上。处于控制中心的计算机对每个种群找到的最优解进行评估,评估将综合考虑蚁群搜索到的最短路径及碰撞风险,然后将评估后的结果通过信息交互机制发送到每个独立的处理单元。每个处理单元上的蚁群根据评估结果进行全局信息素更新。当控制中心达到搜索结束的条件时,控制中心决定机器人移动的下一步,并重新启动下一次的路径搜索,直到机器人达到最终的目标点。

2 控制中心模型

控制中心在整个算法中处于核心位置,控制着整个系统的执行流程。它由传感器获得真实的环境信息进行建模,然后将建模结果发往各个处理单元进行处理,并负责各个处理单元处理过程中信息的交互。

该模型中最为关键的是评估机制,它主要负责在各个处理单元中获得的最短路径和该最短路径与障碍物碰撞风险之间进行决策,指引各处理单元中蚁群的

搜索方向。

假设某个时刻 t ,控制中心获得了 N 个不同处理单元发送的最优解,其中每个最优解是某个处理单元中蚁群迄今为止搜索到的从机器人到子目标点的最短不碰撞路径。对于每一条路径定义该路径的一个评估带状空间ABS,标记与该带状空间相交的 θ 个障碍物集合 $ob = \{o_1, o_2, \dots, o_\theta\}$,如图2所示。

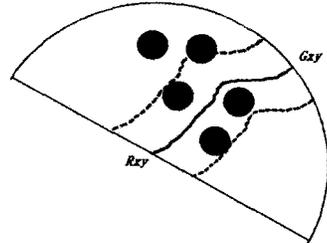


图2 评估带状空间

对于被标记的 $m(m < \theta)$ 个静态障碍物,有:

$$ra^s(t) = \sum_{i=0}^m \frac{1}{D(p_{min}^i(t), p_o^i)}$$

$$= \sum_{i=0}^m \frac{1}{\sqrt{(x_{min}^i(t) - x_o^i)^2 + (y_{min}^i(t) - y_o^i)^2}} \quad (1)$$

其中, $ra^s(t)$ 表示在 t 时刻,每个处理单元上蚁群找到的最优路径与静态障碍物碰撞的风险。其中 $D(p_{min}^i(t), p_o^i)$ 表示最优路径上与静态障碍物最近的点 $p_{min}^i(t)$ 到这个静态障碍物圆心 p_o^i 的直线距离。

对于被标记的 $n(n < \theta)$ 个动态障碍物,有:

$$ra^m(t) = \sum_{i=0}^n \frac{\theta(t)}{\Delta\theta_o(t) + \Delta\theta_i(t)} \quad (2)$$

其中, $ra^m(t)$ 表示在 t 时刻,每个处理单元上蚁群找到的最优路径与动态障碍物碰撞的风险。 $\Delta\theta_o(t)$ 表示在动态障碍物进入传感器范围内的时刻 t 到当前时刻 t 之间角度的变化的最大最小值之差。 $\Delta\theta_i(t)$ 为蚁群在获得最优路径过程中前进方向变化的最大最小值之差。 $\theta(t)$ 为动态障碍物运动方向夹角和蚁群行进方向夹角的交角。

因此,当某个蚁群 a_i 将迄今为止找到的最优解发送到控制中心后,控制中心得到的评估系数为:

$$ra(t) = ra^s(t) + ra^m(t) \quad (3)$$

因此,如果 $ra(t)$ 越大,机器人与障碍物碰撞的风险也就越大。

假设 $ra_1(t) \dots ra_k(t)$ 分别为 k 只蚁群得到的评估系数。相应的 $l_1(t) \dots l_k(t)$ 分别为 k 只蚁群得到的迄今为止得到的最优路径长度,因此,在 t 时刻,机器人选择蚁群 i 找到的最优解的概率为 $P_i(t)$,其定义如下:

$$P_i(t) = 1 - \frac{ra_i^\omega(t) l_i^\sigma(t)}{\sum_{n=1}^k ra_n^\omega(t) l_n^\sigma(t)} \quad (4)$$

其中, ω , σ 分别反映了碰撞风险和路径长度在机器人路径选择中的相对重要性。

其后, 使用轮盘赌方法, 取得概率最大的路径作为机器人迄今为止选择的最优路径。然后根据公式(5)计算相应的信息素更新大小 $\Delta\tau(t)$ 。

$$\Delta\tau(t) = (1 - \zeta) \times \frac{1}{k} \sum_{i=1}^s ra_i(t) + \zeta \times \frac{1}{k} \sum_{i=1}^s l_i(t) \quad (5)$$

其中 ζ 为可变参数, 表征碰撞风险和路径长度对信息素更新的影响。然后将计算得到的 $\Delta\tau(t)$ 和依据轮盘赌得到的最优路径发往每个处理单元中的蚁群来进行全局信息素更新。

3 蚁群模型

对现有的蚁群算法搜索得到的路径中发现, 蚁群在障碍物周围的路径搜索往往决定了蚁群得到的最优路径的好坏^[12], 因此文中引入多蚁群对搜索过程中障碍物周围区域进行加强搜索, 以此引导蚁群的全局搜索, 就能在蚂蚁搜索过程中针对各具体路径和各搜索空间选择合适的信息量, 这无疑提高了解的质量并减少了机器人碰撞的风险。

对真实蚁群社会中的蚂蚁的分工进行模拟, 将算法中的蚂蚁分为两类, 分别为探测蚁、搜索蚁。探测蚁主要对路径中的关键点进行探测, 在本实验中它将对障碍物周围的路径进行探测并留下探测信息素来标记探测的结果, 以便搜索蚁到达该范围后搜索一条更短的不碰撞路径提供辅助信息。搜索蚁的任务是进行全局搜索, 每到一个目标时, 根据侦察素和其它搜索蚁提供的信息素选择下一目标, 直到找到最佳路径为止。

假设每个处理单元中的种群中有 m 只探测蚁, n 只搜索蚁。

对于 m 只搜索蚁, 它将参照探测蚁以及先前蚂蚁留下的信息素值进行全局搜索。搜索规则按照基本蚁群算法进行。

探测蚁则主要用于优化搜索蚁得到的某个障碍物 $ob_i = \{o_1, o_2, \dots, o_\theta\}$ 周围环境空间中的路径, 假设障碍物具有最大影响范围 d_{\max} 和最小安全范围 d_{\min} , 并且蚂蚁由 A 点移动到 B 点, 如图 3 所示。

探测蚁将建立自己的信息素分布并以 A 为蚁穴, B 为食物源进行搜索, 对搜索到的最优路径采用下式对信息素进行更新:

$$\tau(t) = \begin{cases} \gamma(1 - \frac{l_p(t)}{l_s(t)}) + \tau(t-1) & l_p(t) < l_s(t) \\ \tau(t-1) & \text{otherwise} \end{cases} \quad (6)$$

$l_s(t)$, $l_p(t)$ 分别表示探测蚁搜索到的最优路径以及搜索蚁群在该区域中搜索到的路径长度。 γ 为权

值, 取值在 0 ~ 1 之间。

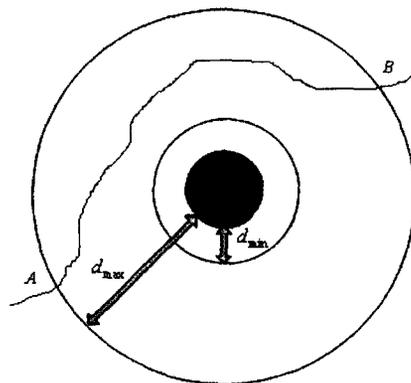


图 3 障碍物周围环境空间

由于蚂蚁选择的随机性^[6], 可能使得到的路径走过一些多余位置, 增加了一些不必要的路径长度。因此在将迄今为止找到的最优解发往控制中心之前, 对最优路径进行拉直操作。具体做法为: 以最优路径上的起点开始, 以一个较小的半径画圆, 将圆与最优路径的交点与起始点用直线连接, 看该直线连接是否与障碍物相交, 如不相交, 则增大一个适当的范围重复上述步骤, 否则, 使用上一次计算得到的不与障碍物相交的点来替代该圆形区域内的最优路径。

4 算法步骤

对于控制中心:

step1: 获得传感器探测到的信息, 创建避障环境, 包括地图环境及障碍物坐标等并对相应的蚁群参数进行初始化。设置迭代次数为 0。

step2: 判断机器人当前是否处于最终目标点。若是, 机器人停止移动。并发送终止信号到各处理单元。否则, 将启动信息和 step1 创建的信息发往各个处理单元进行处理。

step3: 等待处理单元处理, 并在合适的时间间隔内读取各处理单元中迄今为止找到的路径的最优解。

step4: 判断最近几次迭代内获得的最优路径是否相等, 若是, 则认为算法收敛执行 step8, 否则, 执行 step5。

step5: 增加迭代次数, 判断是否达到了最大迭代次数。若是, 执行 step8, 否则, 执行 step6。

step6: 根据第 2 节给出的评估方法进行风险和路径长度的评估, 得到相应的信息素更新值。

step7: 将信息素更新值及评估得到的最优路径发往每个处理单元。然后重新执行 step3。

step8: 发送结束信号到每个处理单元上停止每个处理单元上蚁群的计算。并根据得到的最优路径向前移动一步, 重新执行 step1。

对于每个处理单元上的蚁群: 创建两条进程。

其中一条进程则循环等待控制中心的控制信号,并控制另一条执行蚁群算法进程的执行。

其中控制中心的控制信号包括:

- (1) 终止信号,当收到该信号,程序结束。
- (2) 结束信号,当收到该信号则结束蚁群算法进程的运行。并等待控制中心的下一条指令。
- (3) 获取最优路径信号,进程将取出至今为止找到的最优路径发往控制中心。

(4) 信息素更新信号,进程使用返回的最优路径及相应的信息素更新该蚁群的信息素。

(5) 启动信息,启动蚁群算法进程。

另外一条进程不断地执行蚁群算法,其步骤为:

step1:接收控制中心传递过来的参数并创建机器人避障环境。并初始化迭代次数为0。

step2:创建 $n + m$ 条线程代表 n 只搜索蚂蚁和 m 只探测蚂蚁,算法开始时, m 只蚂蚁探测蚂蚁线程处于休眠等待状态。

step3:判断迭代次数是否为1,如果是,唤醒 m 只处于休眠等待的蚂蚁并执行 step4。否则,直接执行 step4。

step4:处于激活的线程执行一次蚁群路径搜索或探测。

step5:增加迭代次数。

step6:判断迭代次数是否达到给定的值,达到则进行路径拉直操作。

step7:重新执行 step1。

5 实验与分析

5.1 实验分析

为了验证算法可行性,在三台 PC 上使用 Microsoft visual c+进行了仿真,其中控制中心配置为:处理器 intel T2050,内存 512M。两台独立的 PC 配置为:处理器 intel Pentium 4,主频为 2.4GHz,内存 512M。这三台 PC 通过路由器相连。

机器人所处环境为 50×35 的网格空间,分布了 5 个不规则静态障碍物和 2 个圆形动态障碍物,其中蚁群参数设置为: $\alpha = 1, \beta = 2, q_0 = 0.85, \rho = 0.2, \omega = 1.3, \sigma = 1.5, \gamma = 0.5, \zeta = 0.3$ 。

基于分工合作和风险评估的并行蚁群算法 (DRPACO) 最终得到的避障路径如图 4 所示。

其中不规则的黑色区域为静态障碍物,两个圆形区域为动态障碍物,其分别向左右两侧匀速运动,障碍物外部的圆代表了最小安全距离。

5.2 算法比较

为了证明文中提出的算法的有效性,本节对基于文中提出的基于分工合作和风险评估的并行蚁群算法

(DRPACO)、使用 ACO 代替本模型中的蚁群模型得到的并行蚁群算法 (RPACO)、去除控制中心的风险评估,只将每条路径最优的结果返回得到的分工并行蚁群算法 (PACO),以及基本蚁群算法 (ACO) 进行比较,比较结果如图 5 所示。

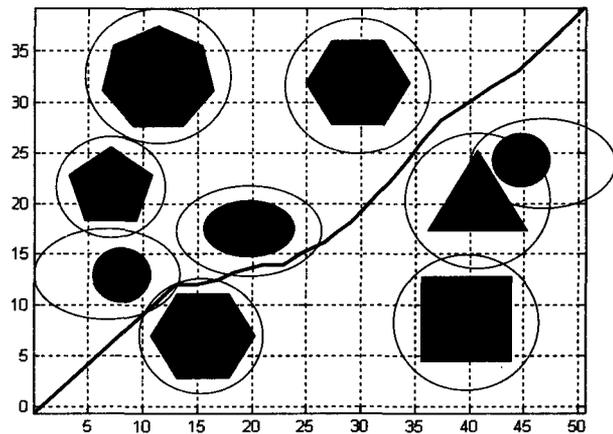


图4 DRPACO 仿真试验结果

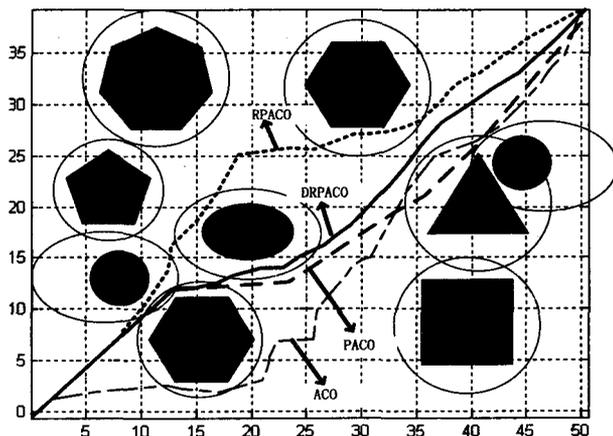


图5 各种算法的比较

从图 5 可以看到,DRPACO 较好地处理了与障碍物的距离关系,避免了碰撞。RPACO 缺少蚁群的全局引导,路径相对 DRPACO 加长。PACO 由于缺少评估机制与障碍物的距离较近,增加了碰撞的风险。ACO 在路径长度上较之前三个算法逊色。

表 1 分别对上述四种情况进行了比较,下表为进行 500 次测试后的比较数据。

表 1 算法比较表

算法	最优解	最优解平均值	平均迭代次数	碰撞次数
DRPACO	68.2431	69.4627	31	0
RPACO	76.4721	79.9462	46	0
PACO	68.6391	70.8576	29	216
ACO	82.2614	85.5735	53	352

从上表可以看到,基于分工和评估的并行蚁群算法 (DRPACO) 得到的路径长度最短且具有较少的收敛

(下转第 18 页)

该软件的开发与传统的软件开发相比,存在一些特殊要求,一些传统的软件可靠性模型并不适用,例如对于弹上软件,真实运行过程中的失效数据难以获取等。通过实际应用的效果,证明该方法在对指导软件开发的过程和质量保证计划的实施,确保软件系统高可靠性要求的时效方面,有良好的效果。

4 结束语

文中主要研究了一种在软件开发过程中的不同阶段采用可靠性预测技术进行软件的可靠性预测的方法,并开发了一套面向开发过程的软件可靠性预测工具。通过实践应用,验证了方法的合理性和科学性。但此方法多基于历史数据的收集和分析,因而下一步的研究重点是收集和分析工程经验和历史数据,提取更精确的评估参数,以提高方法的预测精度和方法的适用性。

参考文献:

[1] 国家标准 GB 6583. 1. 质量管理和质量保证术语[S]. 国家标准局,1986.
 [2] Lyu M F. Handbook of software reliability engineering[M]. USA; McGraw-Hill Companies, 1996.
 [3] 齐治昌. 软件工程[M]. 北京:高等教育出版社,2004.
 [4] 董威,王戟,毛晓光,等. 软件可靠性工程框架和评估

系统实现[J]. 计算机工程与应用,2000,36(10):71-74.

[5] ANSI/AIAA R-013. 推荐的软件可靠性实践[S]. 1993.
 [6] Friedman M A, Tran P Y, Goddard P L. Reliability Techniques for Combined Hardware and Software Systems[R]. Rome Laboratory, Air Force Systems Command, Griffiss Air Force Base, 1992.
 [7] McCall J A, Randell W, Dunham J, et al. Software Reliability, Measurement, and Testing Guidebook for Software Reliability Measurement and Testing[R]. Rome Laboratory, Air Force Systems Command, Griffiss Air Force Base, 1992.
 [8] MIL-HDBK-338B. Military Handbook Electronic Reliability Design Handbook[S]. USA; Department of Defense, 1998.
 [9] Sommerville I. Software Engineering[M]. England: Addison-Wesley Publishers Ltd, 1984.
 [10] Condon E, Cukier M. Applying Software Reliability Models on Security Incidents[C]//18th IEEE International Symposium on Software Reliability Engineering. Toronto: IEEE Computer Society, 2007:150-168.
 [11] Goseva-Popstojanova K, Trivedi K. Architecture-based approach to reliability assessment of software systems[J]. Performance Evaluation, 2001, 45(2-3):179-204.
 [12] Hamlet D, Mason D, Woit D. Theory of software reliability based on components[C]// In: Proc. of the 3rd Int'l. Workshop on Component-Based Software Engineering. Toronto: IEEE Computer Society, 2001:361-370.

(上接第 13 页)

迭代次数,引入的风险评估机制有效地避免了碰撞的发生。

6 结束语

文中提出了一种并行的基于评估和蚁群分工合作的并行移动机器人实施导航方法。该方法利用并行机制加快蚁群的搜索速度,增加机器人搜索的实时性。在保证搜索快速性的同时引入了评估机制,保证搜索的安全性。并在每个处理单元的蚁群中引入分工合作的概念优化蚁群路径。

实验结果表明:该算法具有较好的搜索不碰撞最短路径的能力,能解决机器人在复杂环境中的导航问题。

参考文献:

[1] 段海滨. 蚁群算法原理及其应用[M]. 北京:科学出版社, 2006.
 [2] 毕军,付梦印,张宇河. 一种改进的蚁群算法求解最短路径问题[J]. 计算机工程与应用,2003,39(3):107-109.
 [3] 曾碧,杨宜民. 动态环境下基于蚁群算法的实时路径规划方法[J]. 计算机应用研究,2010,27(3):860-863.

[4] 曾碧. 移动机器人的动态环境建模与路径规划研究[D]. 广州:广东工业大学,2010.
 [5] 凯文,李春葆,秦前清. 基于蚁群算法的最短路径搜索方法研究[J]. 公路交通科技,2006,23(3):128-134.
 [6] Dorigo M, Maniezzo V, Colomni A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man and Cybernetics-part B, 1996, 26(1):29-41.
 [7] 景兴建,王越超,谈大龙. 基于人工协调场的多移动机器人实时协调避碰规划[J]. 控制理论与应用,2004,21(5):757-764.
 [8] Kennedy J, Ebethart R. Particle Swarm Optimization[C]// Proceeding of 1995 IEEE International Conference on Neural Network. [s. l.]:IEEE,1995:1942-1948.
 [9] 朱庆保. 全局未知环境下多机器人运动蚂蚁导航算法[J]. 软件学报,2006,17(9):1890-1898.
 [10] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1):53-66.
 [11] 庄慧忠,杜树新,吴铁军. 机器人路径规划及相关算法研究[J]. 科学通报,2004,20(3):210-215.
 [12] 席裕庚,张纯刚. 一类动态不确定环境下机器人的滚动路径规划[J]. 自动化学报,2002,28(2):161-175.