

# 一种基于概率地图的移动机器人最优期望时间目标搜索

高炳霞, 张波涛<sup>†</sup>, 王 坚, 吴秋轩

(杭州电子科技大学 自动化学院, 杭州 310018)

**摘要:** 在基于概率地图的移动机器人目标搜索规划中, 目标在工作环境中的存在概率通常被设置为服从离散均匀分布, 进而采用路径长度指标优化搜索任务的全局路径. 然而, 真实工作空间中的概率分布绝大多数并不服从均匀分布, 这将导致所获搜索策略并非预期的最短时间. 对此, 根据实际工作环境构建概率测算模型, 并基于该模型构建概率地图, 进而提出一种以预期最短时间为优化指标的机器人目标搜索路径规划方法. 该方法采用分层规划模式, 在上层拓扑地图中进行拓扑点序列规划, 而在下层特征地图中进行拓扑点间局部路径规划. 实验结果表明, 该方法可以显著缩短移动机器人目标搜索的期望时间, 更适用于目标不服从均匀分布的工作环境.

**关键词:** 移动机器人; 目标搜索; 期望时间; 路径规划; 概率地图; 分层规划

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1035

**引用格式:** 高炳霞, 张波涛, 王坚, 等. 一种基于概率地图的移动机器人最优期望时间目标搜索[J]. 控制与决策, 2022, 37(4): 944-952.

## An expected-time optimal target search method based on probabilistic maps

GAO Bing-xia, ZHANG Bo-tao<sup>†</sup>, WANG Jian, WU Qiu-xuan

(School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China)

**Abstract:** In the path planning of target search based on probabilistic maps, the target is usually set to obey discrete uniform distribution in the working space, and then the length of the path is employed as an index to optimize the global path of the search task. However, most of the probability distributions of targets in the real working environment do not satisfy uniform distribution, which will lead to the search path not being the one with the shortest expected time. In order to solve this problem, a probability calculation model is designed according to actual working environments, and based on which, a probability map is designed. With the probability map, an expected-time optimal target search path planning method for robots is proposed, in which, a sequence planning method is used in the upper topology map to obtain the observation point search sequence of the optimal expected time. Finally, a local path planning method is used in the lower feature map to obtain the collision-free path between the observation points. The experimental results show that this method can significantly shorten the expected time of target search, and is more suitable for working situation where the target does not obey uniform distribution.

**Keywords:** mobile robot; target search; expected time; path planning; probabilistic map; hierarchical planning

## 0 引言

移动机器人的目标搜索是指以单个或多个移动机器人在工作环境中对单个或多个目标进行搜索, 以获取其信息或对其操作, 任务执行过程中涉及同步定位与建图(SLAM)、运动控制、路径规划、目标识别、追-逃行为等共性技术<sup>[1-2]</sup>. 移动机器人路径规划可分为全局路径规划、局部路径规划、任务规划等, 移动机器人在目标搜索时的路径规划通常被视为任务规划,

其目的在于根据任务需求按照最短距离、最短时间、最大可靠性等预期目标生成一条最优或次优的可行路径<sup>[3-4]</sup>.

针对移动机器人目标搜索任务的问题, 现有研究通常设置待搜索目标在工作环境内服从均匀分布, 并以最短距离作为指标规划出用于跟踪的路径, 旨在减少获取目标所需的时间, 该方法得到的路径通常接近于时间最优, 属于次优选择<sup>[4-5]</sup>. Sarmiento等<sup>[6]</sup>

收稿日期: 2020-07-26; 录用日期: 2021-02-10.

基金项目: 国家自然科学基金项目(61503108); 浙江省重点研发计划项目(2019C04018).

责任编委: 高会军.

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

发现以最短距离作为指标为目标搜索规划路径,忽略了移动机器人目标搜索过程中的大量不确定性因素,包括目标分布的不确定性、环境的不确定性等,其假设通常与实际情况不符.为使机器人目标搜索策略与现实工作环境具有更高的契合度,文献[6]将概率方法引入到目标搜索中,模仿人类搜索目标的策略构建相应的模型,提出了以期望时间作为机器人目标搜索的优化指标,并认为最优期望时间在很多情况下比最短距离方法更具有合理性,但是该方法的缺陷是假设目标在观测点处服从均匀分布且与可行面积成正比关系,属于非常理想的情况,与机器人的实际工作环境有一定差距. Espinoza等<sup>[7]</sup>在文献[6]的基础上将概率模型由二维平面延伸至三维空间,通过移动机械臂的手眼来扩展机器人视野,减少机器人本体移动,从而构建了面向移动机械臂的最优期望时间目标搜索方法.文献[8]发现当工作空间存在回路时,其在回路中的搜索方向会导致期望时间的差异,并结合概率密度函数的实时更新进一步优化了机器人目标搜索的期望时间.文献[9]提出了以最大可靠性为优化指标的目标搜索方法,尝试以最优安全性作为指标为移动机器人目标搜索规划路径.在基于观测序列规划的目标搜索问题中,客观事实表明了目标在各观测点处的概率并不相等,以最短距离为指标规划出的路径与以最短期望时间生成的路径并不一致.因此,在以节省时间为目标的移动机器人目标搜索任务中,长度最短的路径并非最佳选择.

针对以上问题,本文按照最优期望时间指标进行路径寻优,采用分层式规划策略.尽管蚁群算法在机器人路径规划中的应用不是很理想,但其在旅行商问题等组合优化问题方面具有良好的表现,所以本文上层拓扑序列规划采用蚁群算法搜索期望时间最优的观测点序列.因上层规划只给出节点序列,无法提供适于机器人跟踪的局部路径,故在下层特征地图中采用改进快速随机搜索树(RRT)算法(collision information detection-RRT, CID-RRT)生成易于机器人跟踪的轨迹.该算法在RRT的基础上引入碰撞信息检测和非完整约束,从而使RRT可更快逃脱局部极小值,规划出的路径更适用于非完整约束移动机器人跟踪<sup>[10]</sup>.

## 1 概率测算模型的构建

对于目标搜索策略而言,移动机器人目标搜索采用的方法可分为连续与离散.其中:连续搜索方法是模仿人类在工作环境中的搜索方式,但对全局优化而言效率不高;离散规划的思想是将整个观测区域划

分为若干子观测区域,在其区域中选取观测点,并依次访问.针对两种方法的优劣,一些研究者将两者融合,上层采用全局优化,下层采用局部规划.基于最优期望时间的目标搜索定义如下.

**定义1** 对于移动机器人的工作环境 $S$ ,  $U_k(V_k)$ 表示观测点 $V_k$ 所对应的可视区域,并且该区域应满足

$$U_k(V_k) \in S, \quad (1)$$

观测点应服从

$$V = \{V_k | V_k \in S, k = 1, 2, \dots, n\}. \quad (2)$$

其中: $V$ 是观测点集, $V_k$ 是环境 $S$ 中的第 $k$ 个观测点.基于期望时间的目标搜索路径规划问题是指在 $S$ 中生成观测点 $V_k$ 的最优访问路径.

在移动机器人的实际工作环境中,物体的属性各异,使得其出现在不同区域的概率通常差异非常大,例如:笔记本电脑出现在书房的概率远远高于出现在卫生间的概率,平底锅出现在厨房的概率远远高于出现在卧室的概率.对此,本文引入概率密度因子,以 $U_k(\rho_k, V_k)$ 表示观测点 $V_k$ 所对应的可见区域,其中 $\rho_k$ 是 $U_k(\rho_k, V_k)$ 的密度因子.该密度因子与物体出现在该区域的概率成正相关的关系,表达式如下所示:

$$p_k = \frac{\text{Area}(U_k(\rho_k, V_k))\rho_k}{\sum_{k=1}^n (\text{Area}(U_k(\rho_k, V_k))\rho_k)}. \quad (3)$$

其中: $n$ 表示观测点的数量; $p_k$ 表示目标在观测点 $V_k$ 所覆盖区域存在的概率; $\text{Area}(U_k(\rho_k, V_k))$ 为观测点 $V_k$ 能“看到”的区域,且

$$U_k(\rho_k, V_k) \in S. \quad (4)$$

假设移动机器人匀速运动,且其速度 $v = 1 \text{ m/s}$ ,则时间 $t_k$ 为

$$t_k = L_k/v, \quad (5)$$

其中 $L_k$ 为从起始点开始到第 $k$ 个观测点的路径之和.

根据文献[7]中的离散搜索理论,在机器人的工作环境 $S$ 中,移动机器人搜索目标物体属于离散事件的范畴,其目标的概率分布如下式所示:

$$p(V = V_k) = p_k, k = 1, 2, \dots, n, \quad (6)$$

搜索目标所用的期望时间为

$$E(T) = t_1p_1 + t_2p_2 + \dots + t_np_n = \sum_{k=1}^n t_kp_k. \quad (7)$$

其中: $t_k$ 表示机器人从起点开始到第 $k$ 个观测点的时间, $p_k$ 表示机器人在观测点 $k$ 处存在的概率.累积分布函数 $f(T)$ 表示为

$$E(T) = \int_0^\infty (1 - p(T \leq t))dt = \int_0^\infty (1 - f(T))dt. \quad (8)$$

以4个节点为例,其累积分布函数的示意图如图1所示<sup>[6]</sup>.

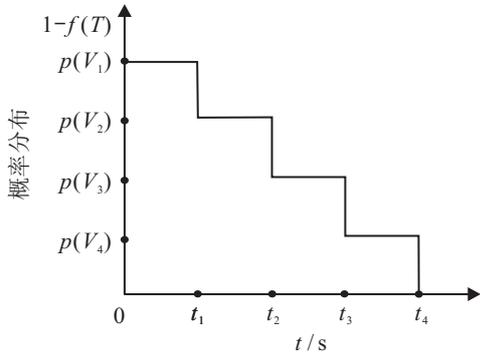


图1 累积分布函数示意图

## 2 基于蚁群算法的上层拓扑序列规划算法

近年来,在基于概率的移动机器人目标搜方法中,国内外研究者通常采用分层规划模式来生成符合指定指标的最优或次最优路径,即:在上层规划中根据任务需求优化节点访问序列,如:文献[11]以贪婪算法搜索拓扑序列,其优点是计算复杂度低,缺点则是很难得到全局最优方案.本文采用蚁群算法(ant colony optimization, ACO)来代替贪婪算法,ACO属于启发式搜索,通过信息素正反馈机制得到全局效果较好的节点序列.

ACO算法属于群体随机优化方法,在环境复杂时易陷入局部极小,故本文通过限定信息素阈值来提高ACO算法的全局优化能力.蚂蚁在寻找最优解的过程中会分泌信息素,蚂蚁选择下一步节点主要依靠信息素及启发式函数,蚂蚁路径构建主要依靠如下公式:

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^c [\eta(i, j)]^d}{\sum_{j \in A(r)} [\tau(i, j)]^c [\eta(i, j)]^d}, & j \in A(r); \\ 0, & j \notin A(r). \end{cases} \quad (9)$$

其中: $j$ 为蚂蚁 $k$ 选取的下一个节点; $\tau(i, j)$ 为节点 $i, j$ 之间的信息素强度; $\eta(i, j)$ 为启发式因素,是节点 $j$ 与目标点之间距离的倒数; $c$ 为信息素影响权重; $d$ 为启发式因素影响权重; $P_k(i, j)$ 为蚂蚁 $k$ 从节点 $i$ 前往节点 $j$ 的概率值; $A$ 为不在蚂蚁 $k$ 禁忌表的节点集合.

随着时间的推移,信息素也会随之逐渐消逝,每只蚂蚁经过一个节点后都通过下式对局部信息素进行更新:

$$\tau(r, s) = (1 - \rho) \cdot \tau(r, s) + \rho \cdot \Delta\tau(r, s). \quad (10)$$

每只蚂蚁完成一次搜索之后,按照下述公式进行全局信息素的更新:

$$\tau(r, s) = (1 - \rho)\tau(r, s) + \rho \sum_{k=1}^m \Delta\tau_{r,s}^k; \quad (11)$$

$$\Delta\tau_{r,s}^k = \begin{cases} Q/L_k, & (r, s) \in R_k; \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

其中: $\rho$ 为信息素挥发因子, $Q$ 为与信息素强度相关的一个常数, $L_k$ 为蚂蚁 $k$ 构建的路径长度, $(r, s) \in R_k$ 表示蚂蚁 $k$ 所经过的边 $(r, s)$ 属于最短路径.

信息素影响因子 $c$ 越大,蚂蚁走之前走过的路的可能性越大,该值过大会导致ACO算法收敛于局部最优. ACO算法中,启发式因素影响权重对算法的影响程度由 $d$ 来决定, $d$ 值越大,说明蚂蚁在该点距离目标的最优期望时间的可能性越大,但易陷入局部最小值.原始算法中, $c, d$ 的值通常为定值,在初始阶段,不能保证可行解中的每一个可行点都受信息素的影响,故本文在ACO算法初始阶段将 $c, d$ 设定为较小的值,从而提高算法的全局搜索能力,在每次迭代后都按照下式对 $c, d$ 进行更新:

$$\begin{cases} c = (c_{\max} - c_0) \frac{N}{D} + c_0, \\ d = (d_{\max} - d_0) \frac{N}{D} + d_0. \end{cases} \quad (13)$$

其中: $c_0, d_0$ 分别表示 $c, d$ 的起始值, $N$ 为当前ACO算法的迭代次数, $D$ 为最大迭代次数.当迭代次数达到最大迭代次数时,便不再对 $c, d$ 值进行更新,记为 $c_{\max}$ 和 $d_{\max}$ .

为了提高ACO算法解的质量,抑制算法收敛于局部最优值,应用最大最小蚂蚁法,在每次信息素更新后,对节点 $i, j$ 之间信息素强度 $\tau_{ij}$ 的值按照下式进行限制:

$$\begin{cases} \tau_{\min}, \tau_{ij} < \tau_{\min}; \\ \tau_{ij}, \tau_{\min} < \tau_{ij} < \tau_{\max}; \\ \tau_{\max}, \tau_{ij} > \tau_{\max}. \end{cases} \quad (14)$$

其中: $\tau_{ij}$ 为节点 $i, j$ 之间的信息素强度, $\tau_{\min}, \tau_{\max}$ 为信息素区间的上下限.将各条路径的信息素集中在区间 $[\tau_{\min}, \tau_{\max}]$ ,这样能够提高其他路径被搜索的可能性,同时还能避免提前收敛于局部最优<sup>[2]</sup>.

由ACO算法得到最优的期望时间序列的步骤如下:

step 1: 参数的初始化,其中蚂蚁的最大迭代次数为 $D$ ,信息素的初始值为常数,信息素增量为0,信息素影响权值 $c$ 和启发式影响权值 $d$ 设为较小的常数.

step 2: 蚂蚁  $i(i = 1, 2, \dots, m)$  从起始位置出发, 并将起始点加入到禁忌表中, 根据式(9)选择下一个节点, 将该节点加到禁忌表.

step 3: 得到寻优期望时间的节点序列, 按照  $E = \sum_{i=1}^n t_i p_i$  评估该节点序列的期望时间.

step 4: 根据期望时间和节点序列, 按照式(11)和(12)更新全局信息素, 再根据式(13)对信息素影响因子和启发式因素影响权重的值进行更新. 完成本轮迭代后, 将禁忌表清空, 若未超出最大迭代次数, 则返回 step 2.

step 5: 当迭代次数 > 最大迭代次数时, 算法结束, 依据  $\min(E = \sum_{i=1}^n t_i p_i)$ , 得到最优期望时间的节点序列.

### 3 基于改进RRT的下层局部路径规划

上层基于拓扑地图所获得的拓扑点序列, 属于路径寻优的概要路线, 并不适用于机器人直接跟踪. 因此, 本文采用改进快速随机搜索树算法进行节点间的路径寻优, 根据局部环境信息, 避开环境中的不可行区域, 生成一条易于跟踪的无碰可行路径.

RRT算法的扩展过程如图2所示, 选出随机树中距离随机生成的点  $V_{rand}$  最近的树节点  $V_{nearest}$ , 并连接这两点.  $V_{nearest}$  沿着该直线移动一个步长  $\zeta$ , 生成新的暂时树节点  $V_{new}$ , 并进行可行性分析. 可行性分析一共返回两个结果: 1) 点  $V_{nearest}$  到点  $V_{new}$  的线段不完全在可行空间内, 说明发生了碰撞, 更新碰撞信息, 并将该方向置为已扩展, 优先考虑距已扩展随机树最远的方向尝试扩展. 2) 点  $V_{nearest}$  到点  $V_{new}$  的路径若满足全局障碍物约束, 则将新生成的树节点  $V_{new}$  加入到随机树中; 若点  $V_{new}$  距目标最终区域的距离小于最大允许距离  $\epsilon$ , 则完成规划<sup>[13]</sup>.

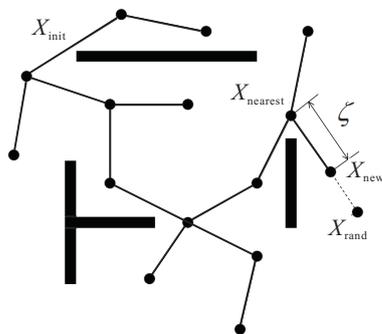


图2 RRT扩展模型

为降低RRT算法的随机性, 引入目标偏好机制

$$f(x) = \begin{cases} 1, & \text{rand}() > \mu; \\ 0, & \text{rand}() < \mu. \end{cases} \quad (15)$$

设置一个阈值  $\mu$ , 如果随机因子大于阈值  $\mu$ , 则在地图上随机选取一个点作为下一个节点的生长方向; 如果小于阈值  $\mu$ , 则将终点作为随机目标点. 这样随机搜索树就能够从起点开始生长, 把终点当作目标, 规划出一条效率更高的路径<sup>[14]</sup>.

为解决目标偏好RRT易陷入局部极小值的问题, 引入碰撞检测机制, 碰撞检测示意图如图3所示. 在地图中随机生成点  $V_{rand}$ , 然后通过截取一个步长确定点  $V_{temp}$ , 因点  $V_{temp}$  与障碍物发生碰撞, 故考虑点  $V_{nearest}$  的其他方向, 优先选择离已扩展随机树最远的方向, 这样能避免同一位置节点重复扩展, 最终选取点  $V_{new}$  作为扩展点加入到随机树中<sup>[8]</sup>.

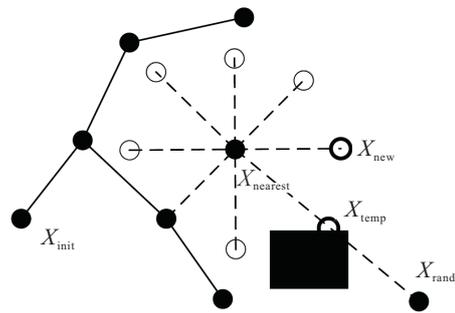


图3 碰撞信息检测模型

扩展过程中每次发生碰撞都会更新随机树上节点的碰撞概率, 如图4所示,  $V_{init}$  是初次扩展的母节点, 其后依据图4的方式逐渐向外生长. 初始化时各个节点均为未发生碰撞的状态, 令每个新节点上的  $a$  为0. 把即将扩展的新节点最临近节点记为  $V_{nearest}$ , 即将扩展的新节点和起始点中间有  $r$  个节点, 当即将

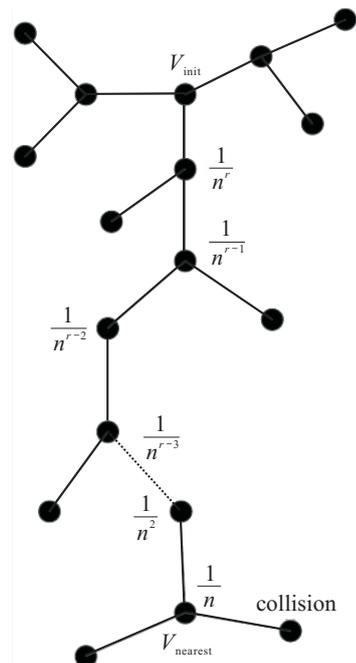


图4 碰撞信息更新模型

扩展的新节点发生碰撞时,  $V_{nearest}$  节点发生碰撞的方向会被置为已扩展, 同时碰撞概率  $a$  会相应增加  $1/n$ , 其中  $n$  为常数,  $V_{nearest}$  节点的母节点的  $a$  会增加  $1/n^2$ . 以此类推,  $V_{nearest}$  的  $r$  阶母节点的  $a$  相应增加  $1/n^{r+1}$ , 一直将  $a$  的信息扩展到根节点. 碰撞概率  $a$  的大小会影响路径的伸展方向,  $a$  值越小, 说明在该扩展方向上碰到障碍物的概率越小. 选择碰撞几率较小的方向来扩展有利于随机树远离障碍物, 同时也能

使随机树快速摆脱局部极小值, 若某个树节点的碰撞概率大于某个阈值或者该树节点的各个方向都是已扩展状态, 则认为该节点的扩展价值不大, 在扩展过程中将不考虑该节点.

基于上层拓扑图的改进ACO算法得到优化后的节点访问序列, 然后在下层规划中采用CID-RRT算法进行局部路径规划, 经过两层优化后得到基于最优期望时间目标搜索路径规划算法如图5所示.

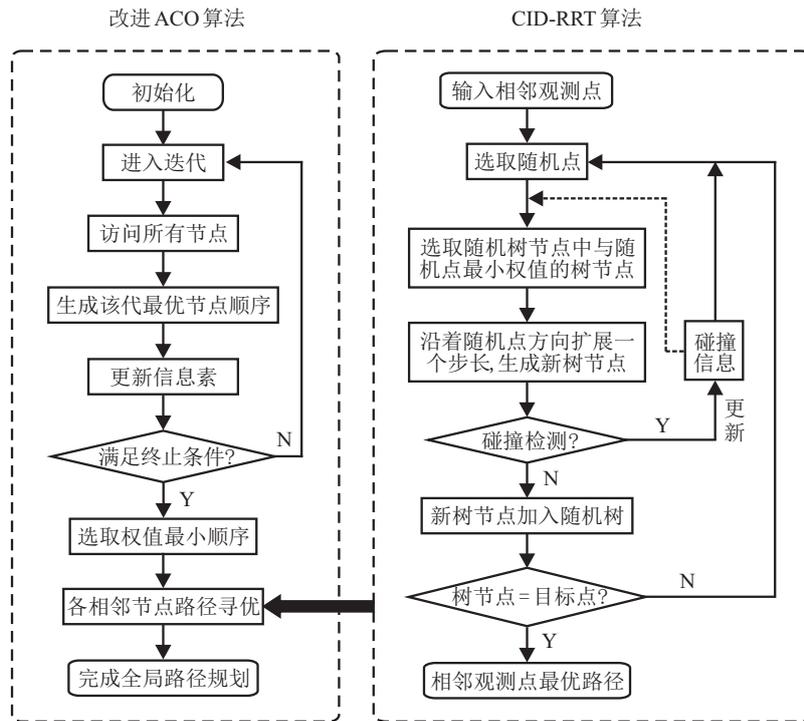


图5 期望时间下的移动机器人目标搜索路径规划

### 4 实验结果分析

为了验证本文所提出观测点的上层序列规划以及下层CID-RRT路径规划能够优化目标搜索的期望时间, 本文在Matlab环境中进行验证, 处理器为Intel Core i5-3230M CPU @2.60 GHz, 内存2 GB. 首先单独验证观测点的序列规划能够缩短目标搜索的期望时间; 然后在不同的地图环境下分别检验ACO算法和贪心算法基于最优期望时间的上层观测点序列规划的优劣, 以及RRT和CID-RRT对点与点之间的路径规划对于移动机器人目标搜索期望时间大小的影响, 并结合算法运行时间进行分析.

优化期望时间所获路径, 与以概率或长度为单目标进行优化有一定区别, 其效果与所采用的地图有一定关联. 本文所提出算法在有的地图中优势较大, 有的则较小, 但是从期望时间来看, 一定优于单独优化概率或长度所生成路径的期望时间.

为了验证以期望时间为指标生成的观测点访问

序列能对目标搜索的期望时间进行优化, 本文在地图环境中选取一个点作为起始点, 记为  $V_0$ , 其后随机生成9个观测点分别记为  $V_1 \sim V_9$ . 为了验证对点的序列优化能够得到最佳的期望时间具有普遍性, 再随机生成9个观测点发现目标的概率. 在地图环境中选取的起始点与随机生成的9个观测点如图6所示, 各个观测点上发现目标的概率如表1所示, ACO算法相关参数如表2所示.

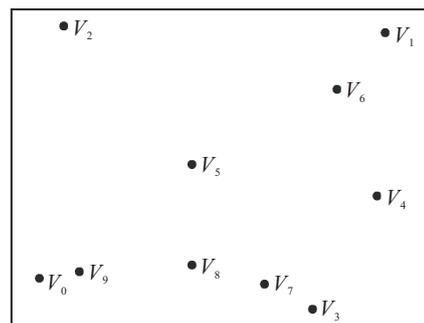


图6 起始点与随机生成9个观测点

表1 起始点与随机生成观测点发现目标的概率分布

观测点	$V_0$	$V_1$	$V_2$	$V_3$	$V_4$
对应概率	0	0.13	0.17	0.03	0.07
观测点	$V_5$	$V_6$	$V_7$	$V_8$	$V_9$
对应概率	0.20	0.11	0.15	0.09	0.05

表2 ACO算法参数表

参数	$c$	$d$	$\rho$	$Q$	$\tau_{min}$	$\tau_{max}$
数值	1	5	0.6	100	0.01	5

贪心算法获得的序列和最优期望时间的序列如图7和图8所示. 贪心算法获得的序列为 $V_0$ - $V_5$ - $V_2$ - $V_7$ - $V_1$ - $V_6$ - $V_8$ - $V_4$ - $V_9$ - $V_3$ ,其期望时间为118.29s;最优期望时间的序列为 $V_0$ - $V_9$ - $V_5$ - $V_8$ - $V_7$ - $V_3$ - $V_4$ - $V_6$ - $V_1$ - $V_2$ ,其期望时间为74.81s. 由此可见,对观测点的序列规划能够显著减小期望时间.

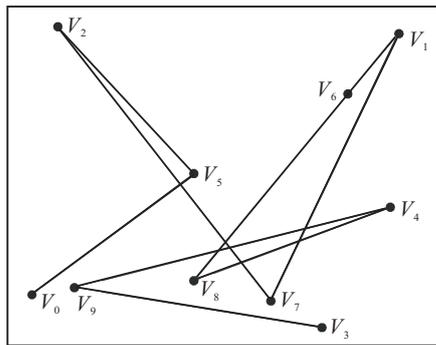


图7 贪心算法得到的目标搜索序列

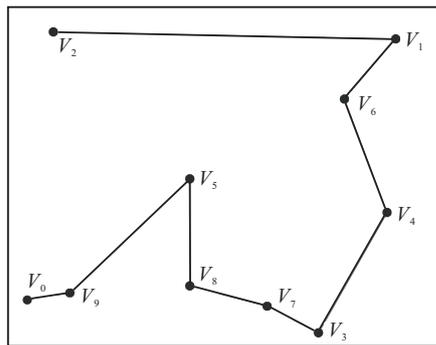


图8 ACO算法得到的目标搜索序列

为了验证对上层观测点的序列规划能够得到更佳期望时间的普遍性,随机生成100组观测点发现目标的概率分布,分别计算贪心算法获得的序列与最优期望时间的序列对应的期望时间,所得结果如图9所示.

为验证本文所提出算法在移动机器人目标搜索任务中的有效性,首先在上层的拓扑序列规划中将本文的ACO算法与贪心算法进行比较;然后在下层

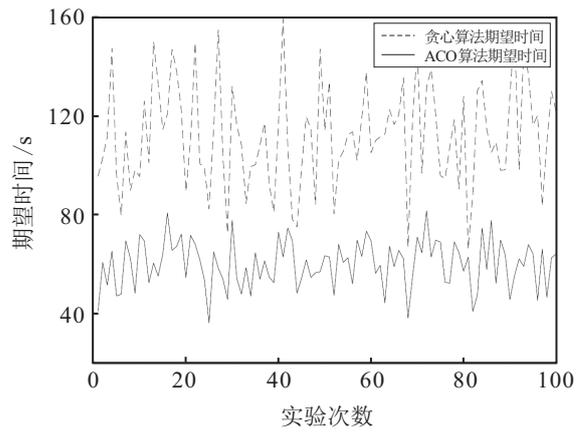


图9 不同概率下两种算法的期望时间

的点对点规划中将原始RRT算法与本文改进的CID-RRT算法进行比较. 环境的不同导致搜索结果存在差异,所以本文采用不同的工作环境来验证所提出两层优化算法的有效性,并搭建工作环境 $S_1$ 和工作环境 $S_2$ ,分别如图10和图11所示.

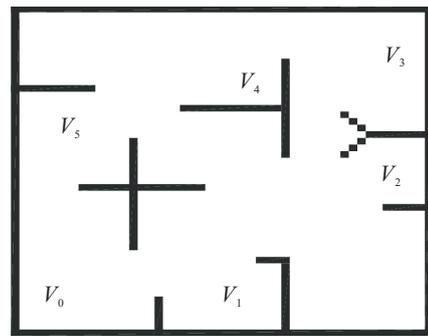


图10 工作环境 $S_1$

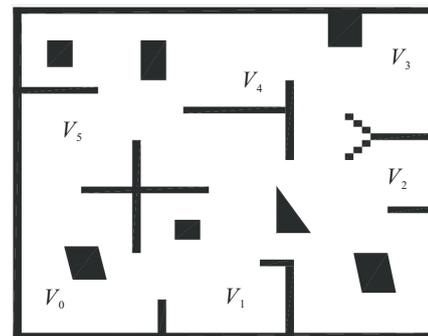


图11 工作环境 $S_2$

对于移动机器人目标搜索任务而言,期望以最少的观测点覆盖待搜索的空间,这样能减少算法处理时间及所需搜索的路径长度,有利于提高移动机器人目标搜索的效率. 面向期望时间的搜索任务举例说明:如图10、图11所示,黑色区域代表障碍物,为不可行区间,其余区域为可行区域.  $V_0$ 为目标搜索的起始点,通过对实验环境进行人工分析,选取5个观测点,分别命名为 $V_1$ 、 $V_2$ 、 $V_3$ 、 $V_4$ 、 $V_5$ ,代表在该点有一定的概率发现搜索的目标.

每个观测点发现目标的概率估算过程:首先将地图环境栅格化,若栅格表征障碍物,则令栅格值为0;若栅格表征地图环境的可行区域,则令其栅格值为1.假设每个观测点都是光源,其发射的光能够穿过栅格值为1的栅格,不能穿过栅格值为0的栅格,故地图环境中能被某个光源照亮的栅格便是该观测点的可见区域;然后结合模拟空间中目标出现的概率及位置的关联人工设置每个观测点的密度因子 $\rho_k$ ;最后通过式(3)得到目标在每个观测点的分布概率,每个点的概率分布如表3所示.搜索任务要求:移动机器人从节点 $V_0$ 开始,按照访问序列依次搜索每个观测点一次,直至最后一个观测点,停止搜索,视为完成目标搜索任务.

表3 环境 $S_1$ 下目标在观测区域的概率分布

观测点	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$
对应概率	0.17	0.21	0.33	0.19	0.1

在 $S_1$ 下分别测试贪心-RRT算法、ACO-RRT算法、贪心-CID-RRT算法和本文所提出算法目标搜索过程中的期望时间的优化情况.在仿真工作环境下,为了减小机器人本身的形状对目标搜索问题的影响,将移动机器人视为质点处理.分别用贪心-RRT算法、ACO-RRT、贪心-CID-RRT算法和本文所提出算法对目标搜索的路径进行规划,生成的路径如图12~图15所示.

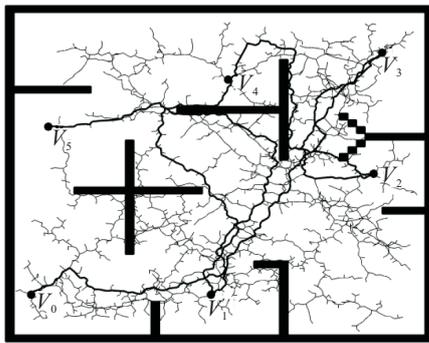


图12  $S_1$ 下基于贪心-RRT的目标搜索路径

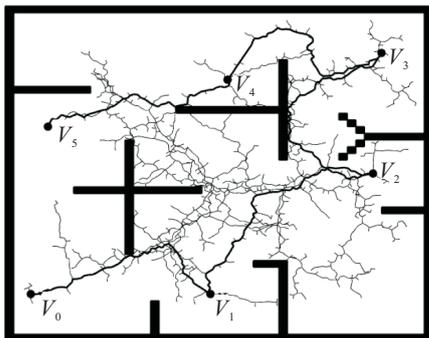


图13  $S_1$ 下基于ACO-RRT的目标搜索路径

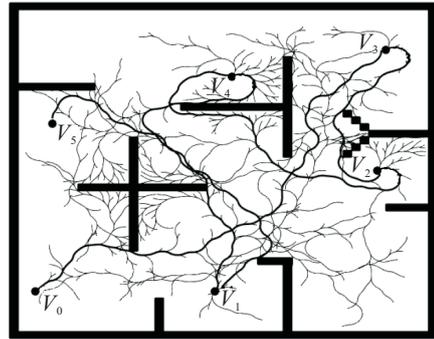


图14  $S_1$ 下基于贪心CID-RRT的目标搜索路径

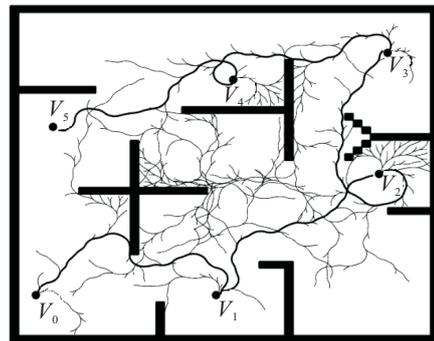


图15  $S_1$ 下基于本文所提出算法的目标搜索路径

从图12~图15可见,上层序列规划中基于贪心算法得到的观测点访问顺序为 $V_0-V_3-V_2-V_4-V_1-V_5$ ,ACO算法生成的观测点访问顺序为 $V_0-V_1-V_2-V_3-V_4-V_5$ ,上层序列规划中只提供了机器人移动的概要路线,为避开实际工作环境的不可行区域,在下层规划中分别采用RRT、CID-RRT生成无碰撞局部路径.将4种算法在环境 $S_1$ 中运行50次,其结果平均值见表4.从运行时间角度看,贪心-RRT算法、ACO-RRT算法、贪心CID-RRT算法、本文算法的运行时间分别为16.07s、9.01s、13.75s、8.20s,由这些数据可见,上层观测点序列规划及下层RRT算法的改进显著缩短了算法的运行时间.从期望时间指标来看,贪心-RRT算法、ACO-RRT算法、贪心CID-RRT算法、本文算法得到的期望时间分别为144.74s、94.64s、131.93s、84.59s,从这些数据可见,经过ACO算法得到最优期望时间观测点访问序列能够极大地减小期望时间,下层局部路径规划对期望时间指标影响

表4 环境 $S_1$ 下4种算法对应的指标

算法	期望时间/s	运行时间/s	提高百分比/%
贪心-RRT	144.74	16.07	—
ACO-RRT	94.64	9.01	34.61
贪心CID-RRT	131.93	13.75	8.85
本文算法	84.59	8.2	41.56

较小.由数据可见,本文所提出算法在期望时间这一指标上有了明显的缩短,较贪心-RRT算法提高了41.56%.

为了验证本文所提出算法在移动机器人不取回搜索这类问题中的有效性,将工作环境复杂化,记为 $S_2$ ,使模型更加贴合于实际环境,并按式(3)对各个观测点的概率进行更新,更新后各个观测点的概率如表5所示.按照上面的步骤重复实验,生成的路径如图16~图19所示.将4种算法分别在环境 $S_2$ 中运行50次,其结果平均值见表6.

表5 环境 $S_2$ 下目标在观测区域的概率分布

观测点	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$
对应概率	0.17	0.14	0.28	0.29	0.12

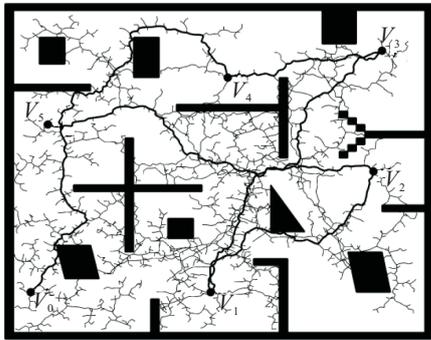


图16  $S_2$ 下基于贪心-RRT的目标搜索路径

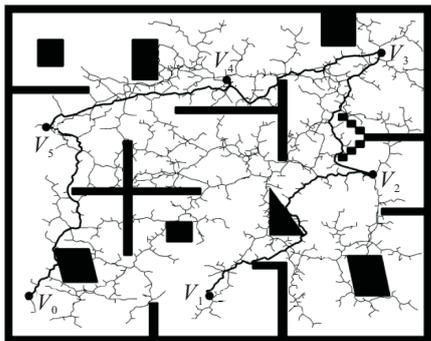


图17  $S_2$ 下基于ACO-RRT的目标搜索路径

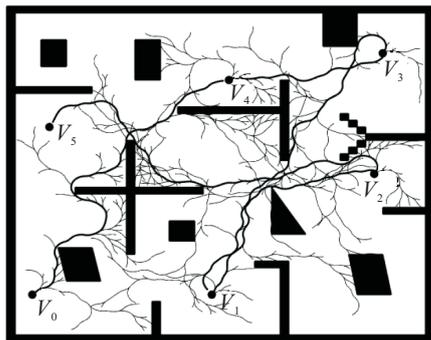


图18  $S_2$ 下基于贪心CID-RRT的目标搜索路径

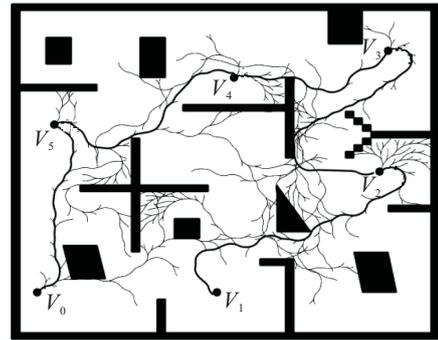


图19  $S_2$ 下基于本文所提出算法的目标搜索路径

表6 环境 $S_2$ 下4种算法对应的指标

算法	期望时间/s	运行时间/s	提高的百分比/%
贪心-RRT	132.62	24.49	—
ACO-RRT	85.46	15.14	35.56
贪心-CID-RRT	126.97	21.06	4.26
本文算法	81.59	12.72	38.48

从图16~图19可见,上层的贪心算法和ACO算法得到的观测点访问序列分别为 $V_0-V_4-V_3-V_1-V_2-V_5$ 、 $V_0-V_5-V_4-V_3-V_2-V_1$ .由表6可见,贪心-RRT算法、ACO-RRT算法、贪心CID-RRT算法、本文算法得到的期望时间分别为132.62s、85.46s、126.97s、81.59s.从这些数据可见,经过ACO算法得到最优期望时间观测点访问序列,能够很大程度地降低期望时间,下层局部路径规划对期望时间指标影响较小.从运行时间角度看,贪心-RRT算法、ACO-RRT算法、贪心CID-RRT算法、本文算法的运行时间分别为24.49s、15.14s、21.06s、12.72s.从这些数据可见,上层序列点的规划明显减低了运行时间.综上所述,本文所提出算法生成的搜索路径显然比贪心-RRT、ACO-CID-RRT等生成的节点访问顺序更加合理,期望时间更小,提高的百分比为38.48.对比表4和表6可见,当观测点的概率发生变化时,期望时间也会随之发生变化.

### 5 结论

针对概率地图下的移动机器人目标搜索问题,本文提出了一种基于最优期望时间的分层式目标搜索路径规划方法.本文引入目标概率分布函数描述目标在环境中的分布特点,在上层拓扑序列规划中采用改进ACO算法优化观测节点的访问序列;在下层的点对点精细路径规划中,以CID-RRT生成易于机器人跟踪的无碰撞局部路径.实验结果显示,本文所提出算法能显著缩短移动机器人搜索获得目标的预期时间,且具有良好的实时性.

## 参考文献(References)

- [1] Meghjani M, Manjanna S, Dudek G. Multi-target rendezvous search[C]. International Conference on Intelligent Robots and Systems. Daejeon, 2016: 2596-2603.
- [2] Schlotfeldt B, Atanasov N, Pappas G J. Maximum information bounds for planning active sensing trajectories[C]. International Conference on Intelligent Robots and Systems. Macau, 2019: 4913-4920.
- [3] Lu W J, Liu D K. A scalable sampling-based optimal path planning approach via search space reduction[J]. IEEE Access, 2019, 7: 153921-153935.
- [4] Konar A, Goswami Chakraborty I, Singh S J, et al. A deterministic improved Q-learning for path planning of a mobile robot[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 43(5): 1141-1153.
- [5] Cheng L P, Liu C X, Yan B. Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot[C]. International Conference on Information and Automation. Hailar, 2014: 695-698.
- [6] Sarmiento A, Murrieta R, Hutchinson S A. An efficient strategy for rapidly finding an object in a polygonal world[C]. International Conference on Intelligent Robots and Systems. Nevada, 2003: 1153-1158.
- [7] Espinoza J, Sarmiento A, Murrieta-Cid R, et al. Motion planning strategy for finding an object with a mobile manipulator in three-dimensional environments[J]. Advanced Robotics, 2011, 25(13/14): 1627-1650.
- [8] Zhang B T, Wang J, Lue Q, et al. An expected-time optimal path planning method for robot target search in uncertain environments[C]. Chinese Control Conference. Wuhan, 2018: 5554-5559.
- [9] Zhang B T, Liu Y, Lu Q, et al. A path planning strategy for searching the most reliable path in uncertain environments[J]. International Journal of Advanced Robotic Systems, 2016, 13(5): 172988141665775.
- [10] 张波涛, 李加东, 刘士荣. 采用碰撞测试和回归机制的非完整约束机器人快速扩展随机树运动规划[J]. 控制理论与应用, 2016, 33(7): 870-878.  
(Zhang B T, Li J D, Liu S R. Rapidly-exploring random trees motion planning for non-holonomic robot with collision-test and regression mechanism[J]. Control Theory & Applications, 2016, 33(7): 870-878.)
- [11] Rashid R, Perumal N, Elamvazuthi I, et al. Mobile robot path planning using ant colony optimization[C]. IEEE International Symposium on Robotics and Manufacturing Automation. Ipoh, 2016: 1-6.
- [12] 曾梦凡, 陈思洋, 张文茜, 等. 利用蚁群算法生成覆盖表: 探索与挖掘[J]. 软件学报, 2016, 27(4): 855-878.  
(Zeng M F, Chen S Y, Zhang W Q, et al. Using ant colony algorithm to generate coverage table: Exploration and mining[J]. Journal of Software, 2016, 27(4): 855-878.)
- [13] Deng H, Xia Z Y, Xiong J. Robotic manipulation planning using dynamic RRT[C]. International Conference on Real-time Computing and Robotics. Angkotr Wat, 2016: 500-504.
- [14] Chen L, Shan Y, Tian W, et al. A fast and efficient double-tree RRT\*-like sampling-based planner applying on mobile robotic systems[J]. IEEE/ASME Transactions on Mechatronics, 2018, 23(6): 2568-2578.

## 作者简介

高炳霞(1995—), 男, 硕士, 从事基于概率拓扑地图的移动机器人目标搜索的研究, E-mail: 825877043@qq.com;

张波涛(1982—), 男, 副教授, 博士, 从事机器人运动规划与控制、机器人感知与决策等研究, E-mail: billow@hdu.edu.cn;

王坚(1980—), 男, 副教授, 博士, 从事机器人控制与规划、机器视觉等研究, E-mail: wangjian@hdu.edu.cn;

吴秋轩(1978—), 男, 副教授, 博士, 从事仿生机器人建模与控制、新能源电力与控制等研究, E-mail: wuqx@hdu.edu.cn.

(责任编辑: 李君玲)