

短学期第二阶段实验指导书

1 路径规划问题描述

首先分析巡检机器人运动的路网。一般情况下机器人允许在油气站场道路上双向行驶，因此可以将机器人运动环境构建为无向图 $G(V, E)$ ，如图 1 所示，其中 $V = [v_1, v_2, \dots, v_n]$ 为道路节点集合， n 为道路节点的数目。巡检机器人的出发点通常为充电房位置（图 1 中节点 s_0 ）。 E 表示无向边集，例如 $E(i, j)$ 表示 v_i 和 v_j 相连的边。图 1 中测量点为 $T_i \in T$ ， $i = 1, 2, \dots, N_T$ ， N_T 表示测量点的数量。权重 $W \in R^{n \times n}$ ，表示机器人通过相应边时经过的距离，例如 w_{ij} 表示机器人从 v_i 运动到 v_j （或反向）经过的距离，满足式（1）所示约束。

$$w_{ij} = \begin{cases} w_{ji}, & i \neq j \\ 0, & i = j \end{cases} \quad (1)$$

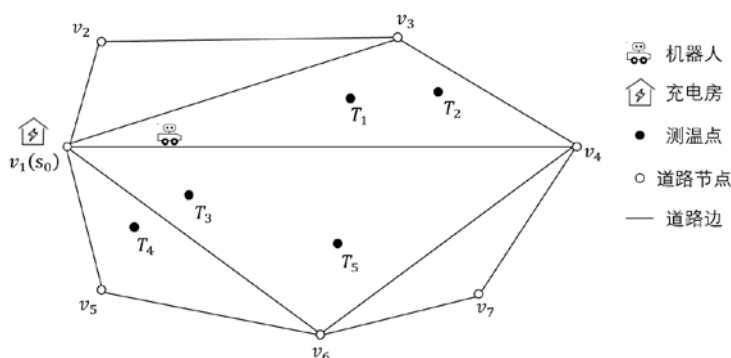


图 1 机器人运动的路网与测量点示意图

为了清晰地表述本文研究的路径规划问题，对机器人工作环境做出以下假设：

（1）取待测目标的几何中心点为测量点 T_i 。忽略变电站道路宽度和坡度的影响，假设道路网络可以完全在二维平面中呈现，即假设各道路节点及各边在竖直轴上的投影坐标均为 0。

（2）为了降低能量消耗，提高任务执行效率，假设机器人在路网中自主选择不重复的巡检路线完成巡检任务，即机器人不会经过同一个道路节点或边两次及两次以上（出发点除外）。

（3）考虑路网和测量点位置分布的随机性，假设任意测量点附近至少存在一条路径或边满足对其进行检测的视距约束。

假设机器人第 k 次停靠实施测量时的位置为 $s_k \in E(i, j)$ ，即处于 $E(i, j)$ 中的某点，并且该次停靠完成测量的目标集为 $T_s(s_k) \in T$ 。 $k = 1, 2, \dots, K$ ， K 表示停靠的总次数，通常 $K \leq N_T$ 。机器人执行巡检任务时首先从充电房出发，在经过节点 $(v'_1, v'_2, \dots, v'_N)$ 的途中依次停靠 s_1, s_2, \dots, s_K 对各目标点进行测量，当完成所有检测任务后返回 s_0 进行充电， $v'_1 = v'_N = s_0$ ，该闭环路径记为 C 。测量的准确性一般取决于测量相机的精度，环境温度，以及图像的采集处理方法。该路径规划问题主要考虑测量数据采集的效率，因此为了获得最小的时间消耗，要求机器人在路网中尽量选择总长度小且停靠次数少的路径，因此，本实验的路径规划问题可以描述为如下多目标约束优化问题 P_1 ：

$$\min_{C, s_1, s_2, \dots, s_K} l(G, T, s_0), K(G, T, s_0) \quad (2)$$

$$s. t. \quad \bigcup_{k=1,2,\dots,K} T_s(s_k) = T \quad (3)$$

$$s_k \in E(i, j), k = 1, 2, \dots, K, v_i \in C, v_j \in C, \forall i, j \quad (4)$$

在问题 P_1 中， C 为 G 的圈，式（2）表示需要对路径的长度 $l(G, T, s_0)$ 进行优化（简记为 l_C ），同时对机器人的停靠次数 $K(G, T, s_0)$ 进行约束。约束式（3）表示所有停靠点的邻域须覆盖全部测量点集。约束式（4）表示所有停靠点必须限制在机器人的巡检路径中。该路径规划问题的目标是获得机器人的巡检路径 C 、各停靠点位置 s_k ，以及各停靠点所对应的测量点集 $T_s(s_k)$ 。

2 巡检机器人路径规划模型

2.1 巡检测量位姿约束

巡检机器人的位姿约束包括检测相机俯仰角约束和最大视距约束。安装相机的云台模型如图3所示，可绕水平轴和竖直轴旋转，其中绕竖直轴旋转范围为 $(-\theta_{max}, \theta_{max})$ ，绕水平轴旋转范围为 $(-\theta_{max}, \theta_{max})$ ，其中 θ_{max} 表示相机受机器人结构和拍摄质量约束的最大俯仰角度， $0 \leq \theta_{max} \leq \pi/2$ 。实际应用中，只需考虑俯仰角上界 $\theta_{max} = 30$ 对巡检任务的影响。

为了便于分析，引入如下定义：测量点约束环 S_{T_i} ，表示机器人对 T_i 进行检测时满足位姿约束的位置集合。设测量仪安装位置离地高度为 H ，测量点 T_i 的坐标为 (x_{T_i}, y_{T_i}, h_i) ， d_i 为机器人停靠点 $s(x_s, y_s, 0)$ 与测量点 T_i 的水平距离，如图2所示。

因此，相机俯仰角度和视距须分别满足如下约束：

$$\arctan\left(\frac{h_i - H}{d_i}\right) \leq \theta_{max} \quad (5)$$

$$\sqrt{d_i^2 + (h_i - H)^2} \leq d_{max} \quad (6)$$

其中， d_{max} 表示相机在满足检测质量要求下的最大视距。式（6）表示机器人必须在相机可视范围内对目标进行测量。其中， d_i 如式（7）所示：

$$d_i = \sqrt{(x_s - x_{T_i})^2 + (y_s - y_{T_i})^2} \quad (7)$$

根据式（5）和式（6），可得测量点的约束环 S_{T_i} 如式（8）所示：

$$S_{T_i} = \left\{ s(x_s, y_s, 0) \left| \frac{h_i - H}{\tan \theta_{max}} \leq \sqrt{(x_s - x_{T_i})^2 + (y_s - y_{T_i})^2} \leq \sqrt{d_{max}^2 - (h_i - H)^2} \right. \right\} \quad (8)$$

上述分析表明，当且仅当机器人位于（8）所示的环境 S_{T_i} 内时才可对 T_i 实现有效测量。

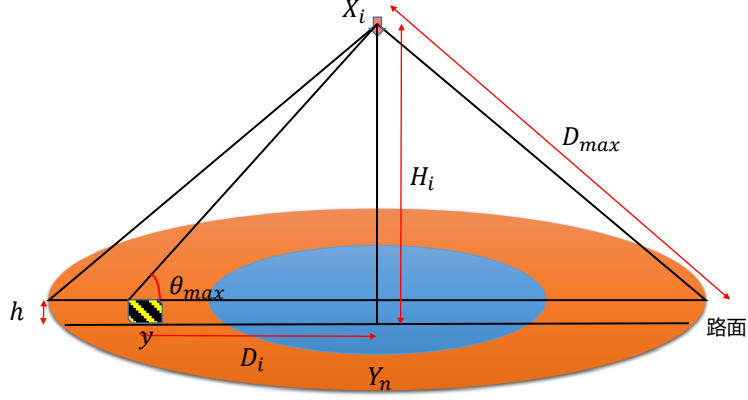


图2 测量点 p_i 的约束环示意图

2.2 巡检路径预处理

油气站场机器人须从充电房 s_0 出发完成测量任务后再返回充电房，记机器人在路网中的巡检路径为 $C = (v'_1, v'_2, \dots, v'_{N'}) \subseteq V$ ，其中， $v'_1 = v'_{N'} = s_0$ 。组成上述路径的边为 $E' = (v'_i, v'_{i+1}) \subseteq E$ ， $i = 1, 2, \dots, N' - 1$ 。该路径总长度由式 (9) 所示：

$$l_C = \sum_{i=1}^{N'-1} w_{v'_i, v'_{i+1}} \quad (9)$$

停靠点 s 可以是机器人巡检路径的边 E' 上的任意一点。为了方便分析，本文通过对路径均匀插值获得离散化路径点。假设相邻两个离散点的间距为 l_L ，组成该路径的离散点集为 L 。机器人在离散化路径中的停靠点由测量点约束环 S_{T_i} 与离散路径 L 的交集确定，如式 (10) 所示：

$$s_k \in \left(\bigcap_{T_i \in T_s(s_k)} S_{T_i} \right) \cap L, k = 1, 2, \dots, K \quad (10)$$

2.3 目标函数

以完成巡检任务总时间最短为目标，路径规划的目标函数如式 (11) 所示：

$$f = \lambda K + \frac{l_C}{v} \quad (11)$$

其中 λ 表示机器人每次停靠测量所消耗的时间常数， λK 为执行 K 次停靠测量所需的时间， $\frac{l_C}{v}$

为机器人在路径中持续运动所需时间， v 是机器人平均运动速度。以机器人在巡检过程中消耗的时间最少为目标，构建如下数学优化模型：

$$\begin{aligned} \min_{C, s_1, s_2, \dots, s_K} & f(G, T, s_0) \\ \text{s. t.} & (1), (4) - (10) \end{aligned}$$

基于上述分析，机器人执行任务的一个例子如图 5 所示。首先，机器人从充电房 s_0 出发沿蓝色箭头所示路径运动，依次停靠在 s_1 对 T_1 和 T_2 测量、停靠在 s_2 对 T_5 测量、停靠在 s_3 对 T_3 和 T_4 测量，最后返回充电房。图 3 中绿色区域是根据式 (8) 画出的各测量点约束环。机

机器人的部分路径被测量环的交集所覆盖，用红色线段表示，这意味机器人停靠在红色路径段上任意一点均可完成对相应目标点的检测。例如，机器人对 T_5 的测量时有 s_2 和 s'_2 两个分离的路段可供选择。

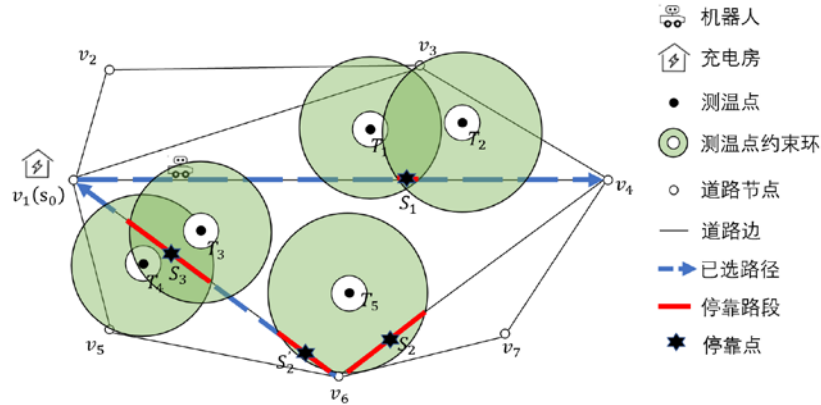


图 3 机器人执行任务示例俯视图

3 求解方法

3.1 A*算法（必做）

A*算法能够通过路径的预测值和实际值在静态环境下快速查找最优路径，路径的预测值和实际值之间的差异越小，则代表搜索目标的速度越快，在相同的应用场景下，该算法比其他搜索算法更具较好的性能和准确度，应用也较为广泛。与Dijkstra算法有着明显的不同，A*算法在启发式策略应用的过程中，搜索更有目的性。A*算法是通过启发函数对现有全局路径规划相关的信息进行计算，寻找搜索代价（或距离最短）最小的节点来进行最优路径规划，A*算法公式如下：

$$f(n) = g(n) + h(n) \quad (3.1)$$

其中， $h(x)$ 是估计代价函数，主要用来反映当前所在节点和搜索目标节点的间距，且满足 $h(x) \geq 0$ 的条件，当 $h(x)$ 值越小表示搜索越接近目标节点。设 $f(n)$ 为当前节点的总代价， $g(n)$ 为起始节点与当前节点的实际代价，当 $f(n) = g(n)$ 时，等价于Dijkstra算法，从初始节点起对所有节点进行搜索，直到找到符合要求的目标节点，根据每个遍历节点代价值选择搜索方向。当 $f(n) = h(n)$ 时，则等价于BFS搜索算法，每次均向着最靠近目标节点的方向搜索，搜索的过程为启发式，但搜索的结果通常不是最优路径。通过对比分析可以得知，Dijkstra搜索算法在实际应用的过程中，搜索范围更加广泛，只要存在解，则无论如何搜索，最终都能够搜索到最终的答案，找到最优路径。但是此种算法需要经过大量计算，搜索效率低。因此A*算法更具优势，具体实现流程如下：

A*算法建立两个列表：open表和close表，open表保存所有已生成而未检验的节点，close表中记录已访问过的节点。其流程为：

- (1) 创建open表和close表，将初始节点放入open表中；
- (2) 对每个节点总代价进行计算，对比分析数值大小，将最小值的节点保存到close表中，表示已经访问过该点；
- (3) 对该节点n所有的邻侧节点进行搜索，实施程序如下：
 - a、若该节点不可通过或者已经在close表中，则忽略该节点。

b、如果该节点已在 open 表中，需要对节点自身相关的 $g(n)$ 进行重复计算，假设当前获取的 $g(n)$ 远远小于之前获取的 $g(n)$ ，就需要重新估算节点，并以此节点为父节点。

c、假设该节点并不在 open 表中，则需要把节点纳入到 open 表。此种情况下节点 n 开始扮演父节点的角色，最后获取该节点自身对应的代价值 $f(n)$ ；

(4) 当目标节点归入到 close 表时，运算结束。通过 close 表获取的节点就被就形成了最短路径，如果最后 open 表里没有节点，但仍未找到目标节点，则表示搜索失败。

该流程可简单理解为：搜索时不停地往两个表中插数据，open 表保证每次访问的节点是其中最优结果，而 closed 表则保证要访问的节点是在之前没有被访问过的。A*算法的流程图如图 3.2 所示。

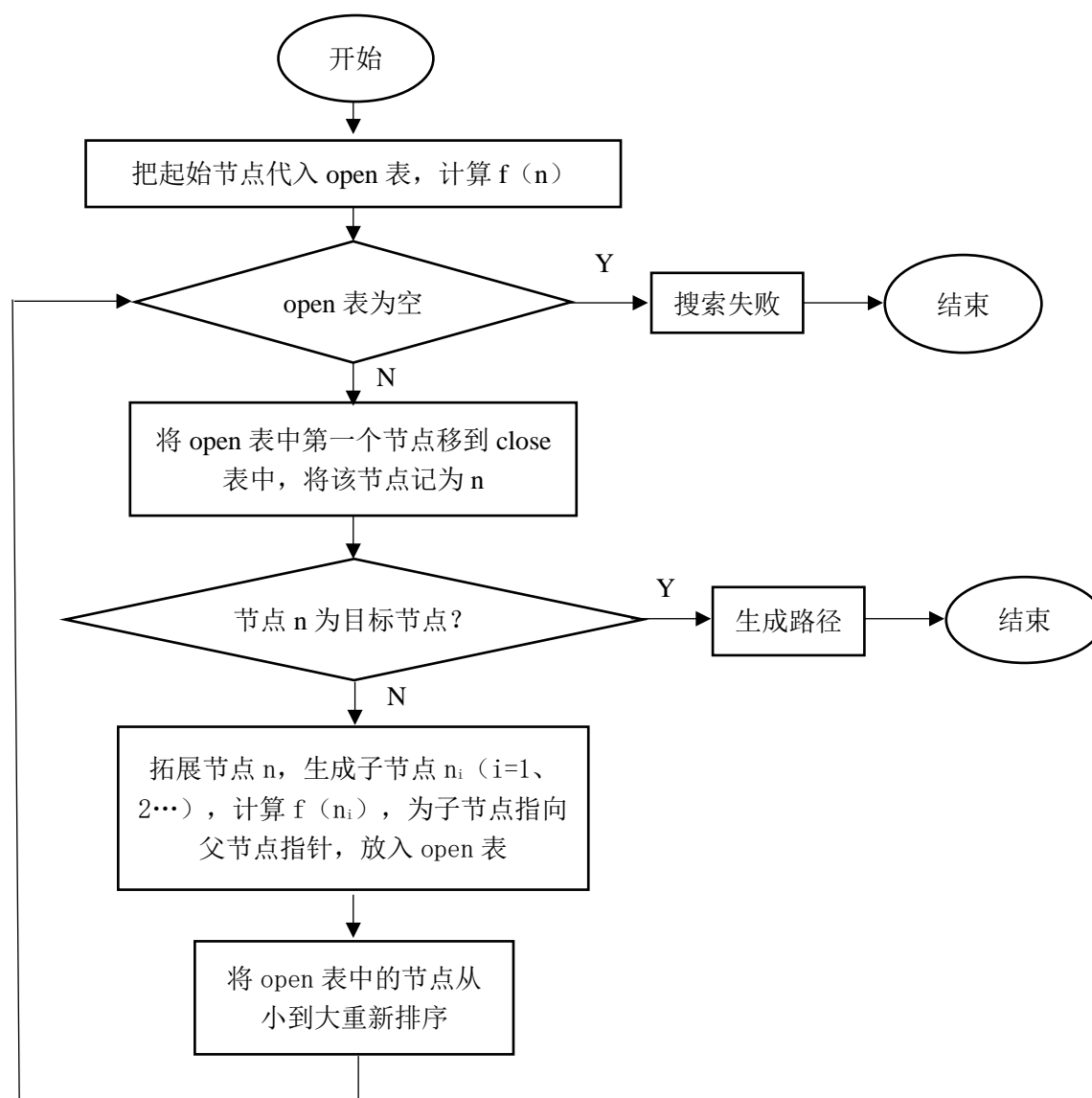


图 4 A*算法流程图

3.2 蚁群算法（选做）

由于前面建立的机器人路径规划模型是一个复杂的多约束非线性优化问题。求解过程中须着重考虑解的构建与更新方法，解的主要部分巡检路径 C 具有路径节点数目不确定的

特点，即解的维数不确定，相关智能算法（包括模拟退火算法，粒子群算法与遗传算法等）难以构建维度不等的解，从而导致这些算法在迭代更新时出现困难。而蚁群算法通过轮盘赌与禁忌表的设计易实现蚂蚁在路网中寻路构建维度不一的初始解，同时易于将解的优劣转化为信息素强弱影响蚂蚁的选择，使路网约束与测量约束贯穿于整个求解过程。

因此本文拟通过改进传统蚁群优化算法进行求解。具体思路如下：首先，以充电房为巡检任务的起点和终点，采用蚁群算法获得一条完整的巡检路径；然后，基于贪婪思想并结合约束条件对测量点进行聚类，计算出蚂蚁路径的停靠次数，再根据目标函数将其换算为时间代价对路网中信息素进行更新。上述过程反复迭代并更新全局最优解，从而求出最优路径、停靠点序列，以及每个停靠点对应的测量点集，求解流程如图 6 所示。

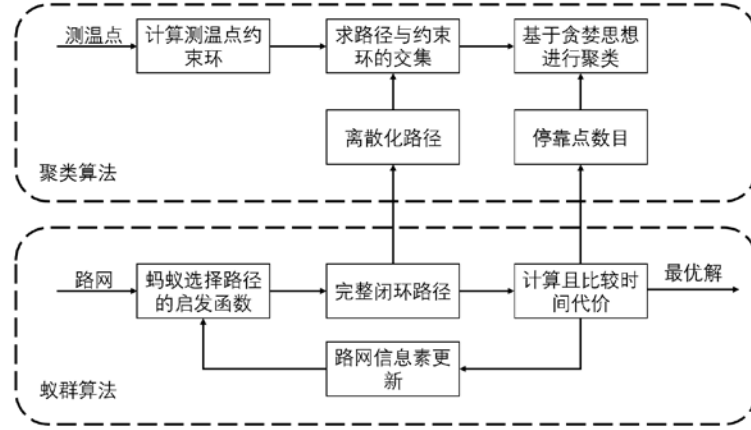


图 6 路径规划求解流程

3.2.1 基于蚁群算法的路径优化

采用蚁群算法对路径优化。设蚁群中蚂蚁的数量为 M ，第 m （ $m = 1, 2, \dots, M$ ）只蚂蚁从路网节点 v_i 转移到 v_j 的概率 p_{ij}^m 如式（12）所示：

$$p_{ij}^m = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{r \in allow_m} [\tau_{ir}(t)]^\alpha [\eta_{ir}(t)]^\beta}, & r \in allow_m \\ 0, & r \notin allow_m \end{cases} \quad (12)$$

其中 $\tau_{ij}(t)$ 为 t 时刻边 $E(i, j)$ 上残留信息素浓度， $\eta_{ij}(t)$ 是启发函数，表示蚂蚁从节点 v_i 转移到 v_j 的期望程度，与节点间距离成反比，如式（13）所示。

$$\eta_{ij}(t) = \frac{1}{w_{ij}} \quad (13)$$

$allow_m$ 表示蚂蚁 m 待访问的节点集合， α 为信息素重要程度因子， β 为启发函数重要程度因子。当所有蚂蚁完成一次循环时，根据式（14）进行迭代。

$$\begin{cases} \tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \\ \Delta\tau_{ij}(t) = \sum_{m=1}^M \Delta\tau_{ij}^m(t) \end{cases} \quad (14)$$

其中 $\rho \in (0, 1)$ 表示信息素的挥发程度， $\Delta\tau_{ij}^m(t)$ 表示蚂蚁 m 在经过的边 $E(i, j)$ 上释放的信息素，如式（15）所示。

$$\Delta\tau_{ij}^m(t) = \begin{cases} \frac{Q}{f_m}, & \text{蚂蚁}m\text{在}E(i,j)\text{上运动} \\ 0, & \text{其他} \end{cases} \quad (15)$$

其中 Q 是一个常量， f_m 是根据式（11）计算的蚂蚁 m 完成任务总时间。 $\Delta\tau_{ij}(t)$ 表示蚁群中所有蚂蚁在边 $E(i,j)$ 上释放的信息素之和。

蚁群算法求解流程如下：

S1：初始化相关参数：蚂蚁数量 M ，信息素重要程度因子 α 、启发函数重要程度因子 β 、信息素的挥发程度 ρ 、信息素释放总量 Q 、最大迭代次数 $iter_max$ 、机器人相机高度 H 、相机俯仰角度最大上限值 θ_{max} 、相机最大视距 d_{max} 、机器人每次停靠测量所消耗的时间常数 λ 、机器人平均运动速度 v 、离散化路径中相邻离散点间距 L_L ；

S2：构建解空间：初始时刻将蚂蚁放在充电房位置，每只蚂蚁根据式（12）依次跳转到随机概率最大的下一个节点，直到该蚂蚁回到出发点或者无可跳转节点。所有蚂蚁执行相同操作。

S3：更新信息素：

①记第 m 只蚂蚁选择的闭环路径为 C^m ，根据式（9）计算其长度 l_C^m （ $m = 1, 2, \dots, M$ ），同时将 C^m 离散化，得到路径点集 L^m ，最后获得 L^m 与第 i 个测量点约束环 S_{T_i} 的交集，记为 $B_i^m = S_{T_i} \cap L^m$ ；

②当 $m = 1$ 时，判断是否存在某个测量点 T_i 使得 $B_i^1 = \emptyset, i = 1, 2, \dots, N_T$ 。若存在，则表示蚂蚁 m 无法完成全部巡检任务，令完成任务总时间 $f_1 = \infty$ ；反之，对路径 C^m 上的所有测量点根据约束进行聚类（将在 3.2 节介绍），得出类别总数 K^1 ，将其代入式(11)计算巡检时间代价 f_1 。依次计算当 $m = 2, 3, \dots, M$ 时所对应的 f_2, f_3, \dots, f_M 值，根据式（14）和式（15）更新信息素。信息素更新流程如图 7 所示；

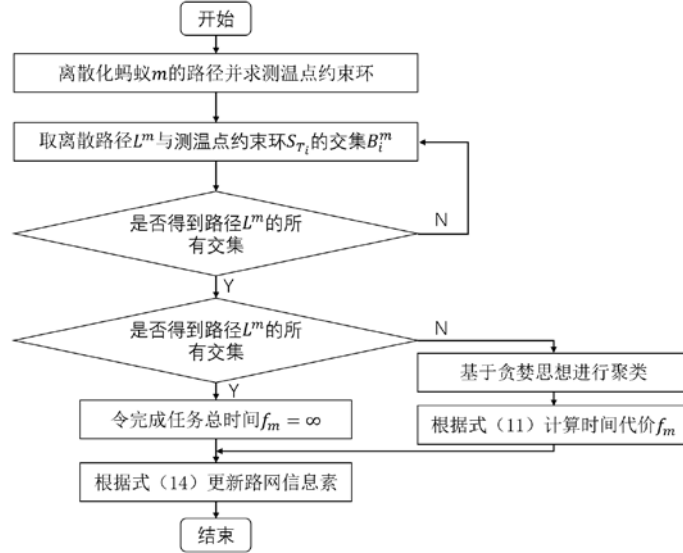


图 7 路网中信息素更新流程

S4：判断蚁群是否终止迭代：计算当前迭代中所有蚂蚁完成巡检任务的时间消耗，同时与以前迭代中获得的全局最优解比较，当存在更小的时间代价时，则更新全局最优解。然后，判断迭代次数是否达到最大值，若未达到，则返回第 S2 步，若达到则停止计算，输出全局最优解对应的巡检路径 C 、各停靠点位置 s_k ，以及各停靠点所对应的测量点集 $T_s(s_k)$ 。

3.2.2 基于贪婪思想的测量点聚类

在蚂蚁获得一条完整的巡检路径后，对测量点聚类以获得机器人在该路径下的测量停靠方案。聚类获得的类别数目便是机器人的停靠次数。基于贪婪思想将满足测量约束的测量点聚为一类，通过迭代获得聚类结果。聚类算法流程如下：

S1: 构建待聚类的测量点初始集合 $A = T$ ，求蚂蚁 m 的离散化路径 L^m 与第 i 个测量点约束环 S_{T_i} 的交集 B_i^m ， $i = 1, 2, \dots, N_T$ ；

S2: 由式 (16) 计算路径点集 $B_{i,j}^m$ ， $j > i$ 且测量点 $T_j \in A$ 。集合 $B_{i,j}^m$ 中离散点的数目 $N_{i,j}^m = |B_{i,j}^m|$ 代表了 T_i 与 T_j 的相近程度。将所有 $N_{i,j}^m$ 从大到小排序，得 $N_{i,j_1}^m \geq N_{i,j_2}^m \geq \dots \geq N_{i,j_Z}^m$ ，其中 $Z = |A| - 1$ ；

$$B_{i,j}^m = B_i^m \cap B_j^m \quad (16)$$

S3: 如果 $N_{i,j_1}^m = 0$ ，表示 T_i 与其它目标点没有共同可行的停靠路径点，则将 T_i 单独归为一类，即 $T_s^m(s_k) = \{T_i\}$ ，然后执行 S5 计算蚂蚁 m 剩余停靠点，否则，执行 S4；

S4: 如果 $N_{i,j_1}^m > 0$ ，则路径点集 $B_{i,j_1}^m \neq \emptyset$ 。定义如下可行的停靠路径集合 $D_{i,j_r+1}^m = D_{i,j_r}^m \cap B_{i,j_r+1}^m$ ，其中 $D_{i,j_1}^m = B_{i,j_1}^m$ ， $r = 1, 2, \dots, Z$ 。易求得：

$$r^* = \max(r), \quad s.t. D_{i,j_r}^m \neq \emptyset$$

则得到停靠点 s_k^m 可完成测量的目标集 $T_s^m(s_k) = \{T_i, T_{j_1}, \dots, T_{j_{r^*}}\}$ ，式 (17) 为相应停靠区域集合表达式：

$$B_{T_s^m(s_k)}^m = \left(\bigcap_{T_i \in T_s^m(s_k)} S_{T_i} \right) \cap L^m, i = 1, 2, \dots, K \quad (17)$$

于是根据式 (10)，停靠点 s_k^m 从停靠区域集 $B_{T_s^m(s_k)}^m$ 中任意选择即可；

S5: 从 A 中剔除能够完成检测的目标集 $T_s^m(s_k)$ ，即：

$$A = A - T_s^m(s_k)$$

从 A 中选择下一个测量点执行 S2，直到 A 为空集时，得到蚂蚁 m 的路径对应的停靠点位置 $s_1^m, s_2^m, \dots, s_K^m$ 与其所对应的测量点集 $T_s^m(s_1), T_s^m(s_2), \dots, T_s^m(s_K)$ ；

S6: 最后根据机器人闭环路径 C 中道路节点的先后顺序调整 s_k^m 的顺序，得到最终的 $s_1^m, s_2^m, \dots, s_K^m$ 与 $T_s^m(s_1), T_s^m(s_2), \dots, T_s^m(s_K)$ ，避免机器人在此路径中往复运动。

该聚类算法的核心是将离散路径与测量点约束环的交集作为基本处理单元，以各个单元之间的交集大小来表征当前路径下目标点之间的相近程度，从而来确定是否为同一类。

4 实验数据

实际路网实验基于油气站场虚拟环境与实地勘测，如图 7 中所示。从位于左上角充电房（蓝色六角星 ★）开始，经过指定的检测目标仪表点（黑色圆点 ●）后，回到充电房的过程。其中，起点编号为 0，标号 1~26 代表 26 个指针仪表点。路径表示为两个邻接顶点编号，例如，路径 (27, 28)，(66, 39) 等。

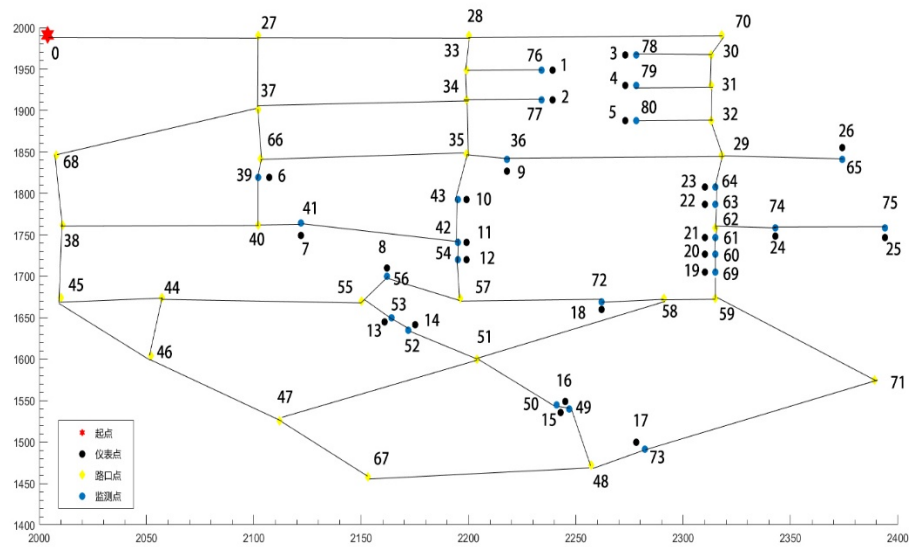


图 7 实际油气站场路网

两个顶点间路径长度可以视为求欧式距离，每个顶点坐标如表 1 所示。

表 1 顶点坐标及类型

顶点编号	X	Y	类型
0	2004	1991	起始点
1	2239	1949	仪表 1
2	2239	1913	仪表 2
3	2270	1967	仪表 3
4	2273	1930	仪表 4
5	2273	1888	仪表 5
6	2107	1819	仪表 6
7	2122	1749	仪表 7
8	2162	1710	仪表 8
9	2218	1827	仪表 9
10	2199	1793	仪表 10
11	2199	1741	仪表 11
12	2199	1720	仪表 12
13	2161	1645	仪表 13
14	2175	1642	仪表 14
15	2243	1536	仪表 15
16	2245	1549	仪表 16
17	2281	1492	仪表 17
18	2262	1667	仪表 18
19	2313	1705	仪表 19
20	2313	1727	仪表 20
21	2313	1747	仪表 21
22	2313	1787	仪表 22

23	2313	1808	仪表 23
24	2343	1748	仪表 24
25	2394	1747	仪表 25
26	2374	1847	仪表 26
27	2102	1990	路网节点
28	2200	1990	路网节点
29	2318	1846	路网节点
30	2313	1967	路网节点
31	2313	1930	路网节点
32	2313	1888	路网节点
33	2199	1949	路网节点
34	2199	1913	路网节点
35	2199	1848	路网节点
36	2218	1841	仪表 9 建议观测点
37	2102	1901	路网节点
38	2011	1762	路网节点
39	2102	1819	仪表 6 建议观测点
40	2102	1762	路网节点
41	2122	1764	仪表 7 建议观测点
42	2198	1741	仪表 11 建议观测点
43	2198	1793	仪表 10 建议观测点
44	2057	1674	路网节点
45	2010	1674	路网节点
46	2052	1604	路网节点
47	2112	1526	路网节点
48	2257	1472	路网节点
49	2244	1548	仪表 16 建议观测点
50	2245	1538	仪表 15 建议观测点
51	2204	1600	路网节点
52	2173	1640	仪表 14 建议观测点
53	2166	1650	仪表 13 建议观测点
54	2198	1720	仪表 12 建议观测点
55	2163	1670	路网节点
56	2162	1670	仪表 8 建议观测点
57	2196	1673	路网节点
58	2291	1673	路网节点
59	2317	1714	路网节点
60	2315	1727	仪表 20 建议观测点
61	2315	1747	仪表 21 建议观测点
62	2315	1753	路网节点
63	2315	1787	仪表 22 建议观测点
64	2315	1808	仪表 23 建议观测点
65	2374	1846	仪表 26 建议观测点

66	2103	1842	路网节点
67	2153	1458	路网节点
68	2008	1846	路网节点
69	2315	1705	仪表 19 建议观测点
70	2318	1991	路网节点
71	2389	1575	路网节点
72	2262	1669	仪表 18 建议观测点
73	2282	1491	仪表 17 建议观测点
74	2343	1751	仪表 24 建议观测点
75	2394	1754	仪表 25 建议观测点
76	2235	1949	仪表 1 建议观测点
77	2235	1913	仪表 2 建议观测点
78	2274	1967	仪表 3 建议观测点
79	2277	1930	仪表 4 建议观测点
80	2277	1888	仪表 5 建议观测点

4.1 油气站场环境建模

路径规划时使用栅格法相当于将移动机器人的工作环境模拟成为栅格地图，从而对移动机器人工作空间进行数学模型构建。在栅格地图中，移动机器人工作空间被转化为平面二维数组，且障碍物的位置与体积一经确定，将不会发生变化。根据四足机器人的大小，在栅格地图的构建中，先设定单个网格的边长 R ，边长被固定后，每一个网格的面积即为 $W = R^2$ ， W 是一个网格，则此时 W 就分割成了面积相同的正方形栅格工作空间。同时由于在每一个栅格上记录着机器人的移动情况以及障碍信息，因此栅格的属性也被确定。假设障碍物存在于某一栅格内，则此栅格被定义为障碍栅格，若任一栅格之中没有任何的障碍物，则这样的栅格就是自由栅格，此时机器人就能够通过栅格。当栅格之中存在障碍物时，不管是否障碍物占据整个栅格，此时都应该依据障碍物的栅格来进行区分，对起始位置的点 S 当作机器人的起点，将目标位置所在的点 D 定义为移动机器人的目标栅格。

如图 8 所示，以油气站场内道路宽度为边长的正方形栅格作为最小单位，建立了 $M \times N$ 单位长度大小的环境地图。

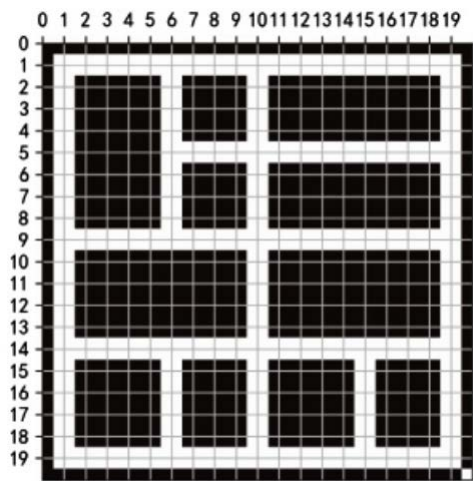


图 8 油气站场栅格图示意图

结合油气站场环境的实际情况，设定四足巡检机器人的移动方向为前、后、左、右、右前、右后、左后、左前，8 个方向，不能跨越障碍物，不能沿栅格对角线方向前进，如图 9 所示。

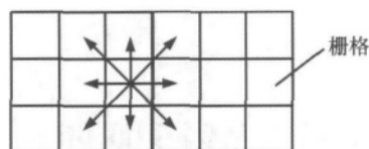


图 9 巡检机器人可移动方向示意图

图 10 位四足机器人的基本信息，在巡检过程中，路径的宽度要大于四足机器人的体宽。

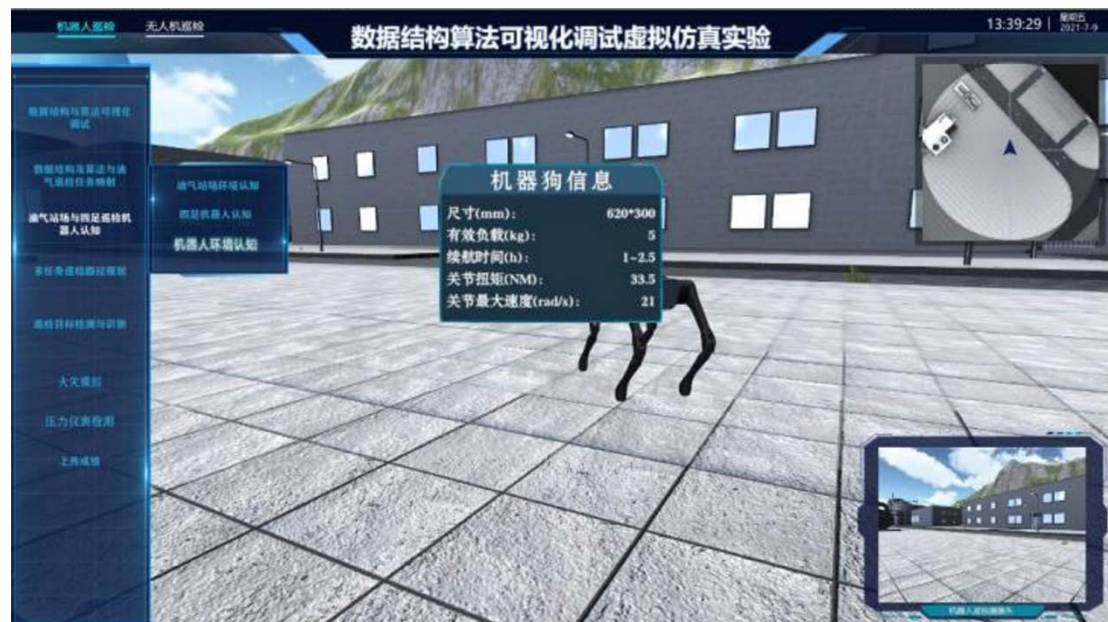


图 10 巡检机器人尺寸信息

4.2 栅格尺寸的选取

利用栅格法的实质应该是路径信息的离散化，所以在地图中的信息应该与实际路径存在一定的差别，所以难免会存在一些误差，而这种误差通常是由于地图的分辨率而导致，因而栅格可以说是通过分辨率来确定的。基于此，可以把栅格尺寸当作是评价一个地图质量高低的标尺。当栅格的尺寸比较小时，则地图的精度就会提升，因而误差就会降低，地图表示与实际路径差别就会很小。如果环境为一个大型的场地，随着栅格的数量的增加，且栅格变小，则网格的数目就会随之增多，所占的内存也会相应的增加，这就会导致搜索速度变慢，从而增加了路径搜索的时间。反之如果每个栅格的大小被设计的偏大，会影响最终结果的精确度。基于此，如何合理的选取栅格的大小将是一个关键问题。综合实验区域等客观因素，确定栅格地图的行列数： $M = R/RL$ ， $N = M$ 。栅格尺寸与数量的合理选取也使对精度的影响降到了最低。

4.3 栅格尺寸的标识

如果应用栅格法描述环境空间，栅格的标识就是必不可少的，直角坐标系法和序号法已经成为常用的栅格标识的方法。采用直角坐标系法，对处在工作环境中的左上角 W 设定为坐标原点， X 代表是水平位置，同时定义为向右为正，呈现越向右值越大的趋势。 Y 类似，向上为正方向，同时随着向上方向的延伸值越大，每一个栅格都可以通过固定的 X 与 Y 来进行确定。所以直角坐标 (X,Y) 可以用来表示任意栅格，其坐标即为栅格的左上角坐标。

栅格的信息编码当通过栅格法进行区域处理建立起环境模型后，就可以进一步进行对应的编码工作。由于栅格与数组存在对应的联系，因而矩阵进行行列分割能够实现栅格的存储。数组矩阵之中，栅格的坐标是结构体的索引，任何的栅格的信息都能够通过数组进行表示，其中结构体与地图中栅格进行对应。栅格的坐标值在直角坐标系中被 X 和 Y 表示，障碍物的选择标记被赋予 0 与 1 两个值。如果是 0，则意味该栅格为自由栅格，可以通行；如果是 1，则意味该栅格为障碍栅格，不能通行。在此基础上，搜索过程中栅格地图就会显示出来。根据具体路径信息，在栅格地图上添加静态障碍物，障碍物经过栅格化处理后以二维数组形式存储在工作空间。本章中选取两组不同栅格密度的工作空间，工作空间中静态障碍物用红色圆圈表示。

5 任务描述

5.1 Python 实现 A*路径规划算法（必做）

（1）基于图 7 中的数据，按照图 8 的形式制作四足站场模型。使用 Python 语言实现 A*路径算法，规划路径需要经过 3、21、24、17、8、6、13、18、7、25 号仪表点（黑色圆点）进行巡检。

（2）将 A*算法代码输入 <http://pythontutor.com/visualize.html#mode=edit>，进行可视化调试，给出关键步骤（不少于 5 步）的代码执行过程中堆结构的输出结果，如图 11 所示。



图 11 算法中执行中堆结构结果

- (3) 路径规划完成，登录 dsavd.cup.edu.cn，将路径导入虚拟仿真系统进行验证。
- (4) 将四组机器人视角拍照的图片（步骤 1 中 10 个位置仪表）送入第一阶段搭建的仪表检测模型进行目标检测实验，并将识别结果写入实验报告。

5.2 Python 实现蚁群路径规划算法（选做）

5.3 油气站场巡检系统方案设计（必做）

结合第一阶段的深度学习仪表检测方法，设计油气站场实际巡检系统方案，包括：路径规划、指针仪表目标检测、仪表读数识别等功能并给出阶段算法的设计理由。