



第三讲 路径规划

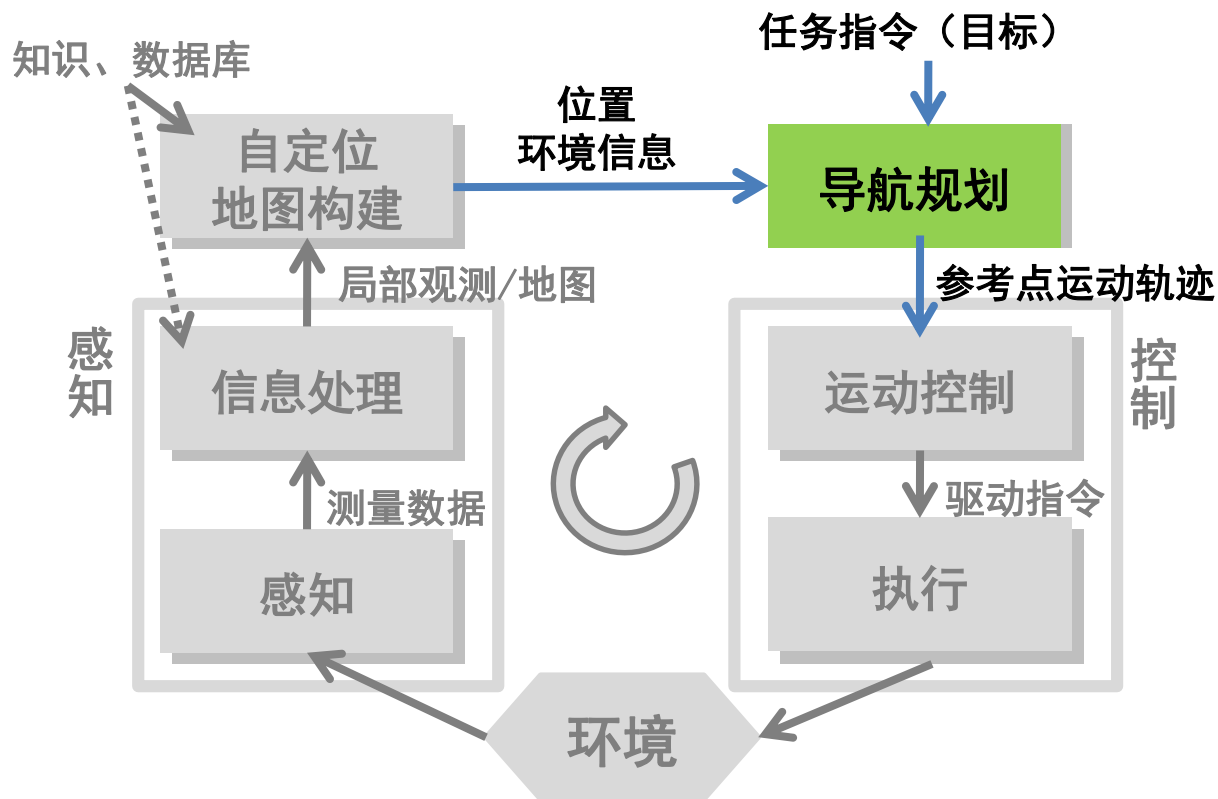
王越

浙江大学 控制科学与工程学院

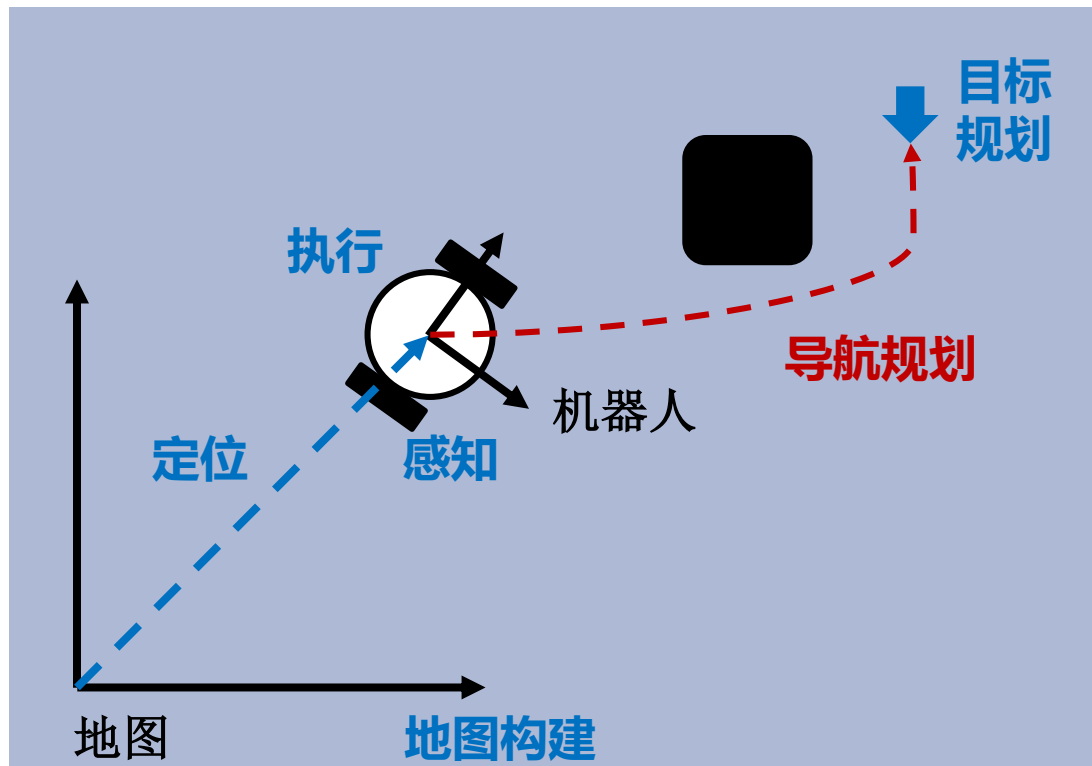


导航规划简介

自主移动机器人一般架构



导航规划



在给定环境的全局或局部知识以及一个或者一系列目标位置的情况下,使机器人能够根据知识和传感器感知信息高效可靠地到达目标位置

导航规划类型

- 固定路径导引：
 - 有人工标识导引
- 无轨导航：
 - 有人工标识导引的无固定路径（无轨）导航
 - 无标识导引的自然无轨导航



导航规划类型1：有人工标识导引的固定路径导航



磁条导航



磁感应线导航



磁钉导航



二维码导航



AGV: Automatic Guided Vehicle 自动导引车

- 优点：技术成熟、稳定可靠、价格优惠
- 缺点：需要施工和维护、路线无法调整

导航规划类型2：有人工标识导引的无轨导航



激光反射板导航



- 优点：技术成熟、路径可调
- 缺点：需要施工和维护、价格昂贵



导航规划类型3：无人工标识导引的无轨导航



自然导航

- 优点：无需施工、路径可调、精确定位、室内外通用
- 缺点：算法复杂，环境变化影响定位可靠性和稳定性



导航规划问题

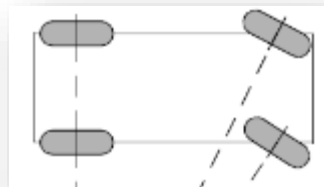
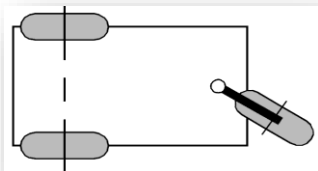
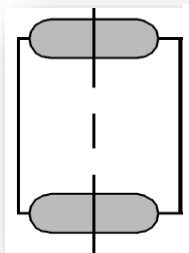
在给定环境的全局或局部知识以及一个或者一系列目标位置的条件下，使机器人能够根据知识和传感器感知信息高效可靠地到达目标位置

- 环境几何约束
- 机器人执行约束



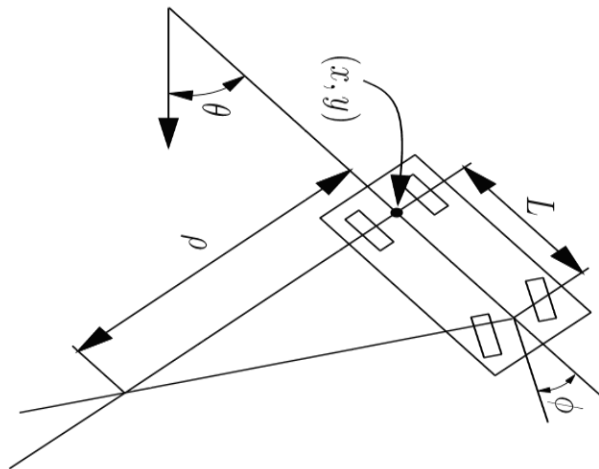
非完整机器人 NON-HOLOMIC

- 存在至少一个非完整运动学约束

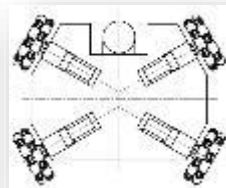
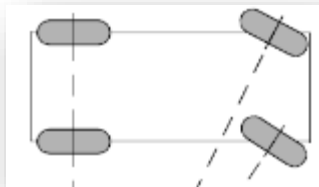
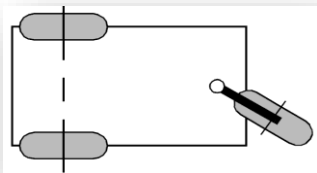
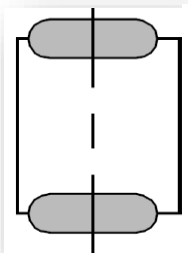


其他运动学约束

- 最大转弯半径
- 最大速度、最大加速度

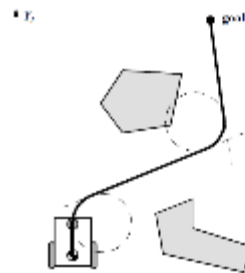
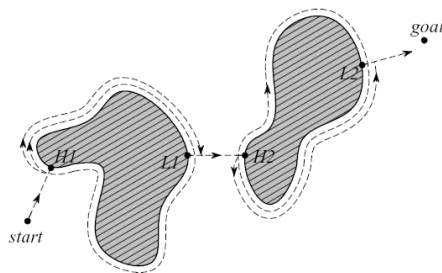
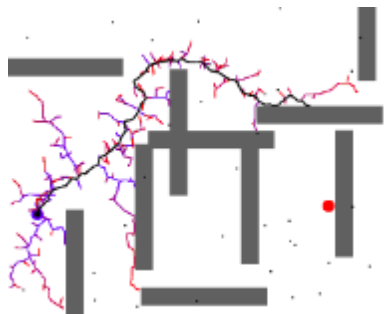


移动机器人形态丰富，运动学模型和约束各不相同

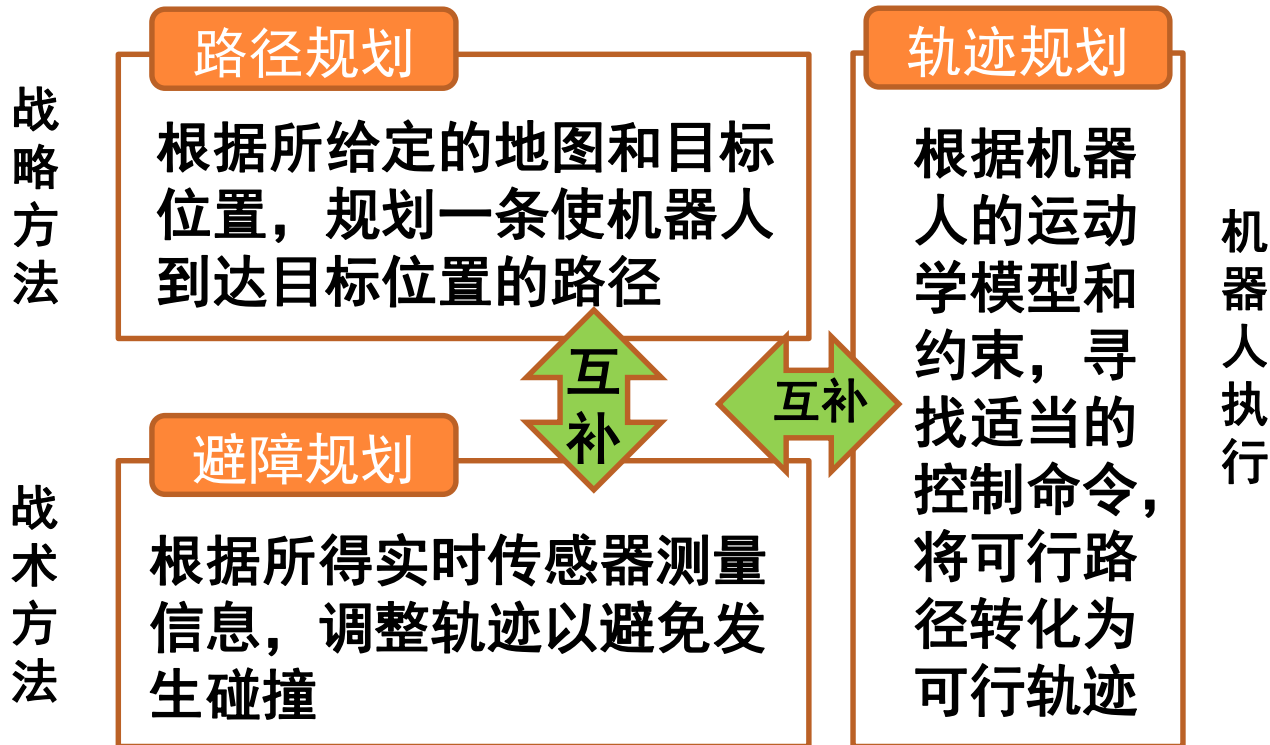


导航规划的主要研究内容

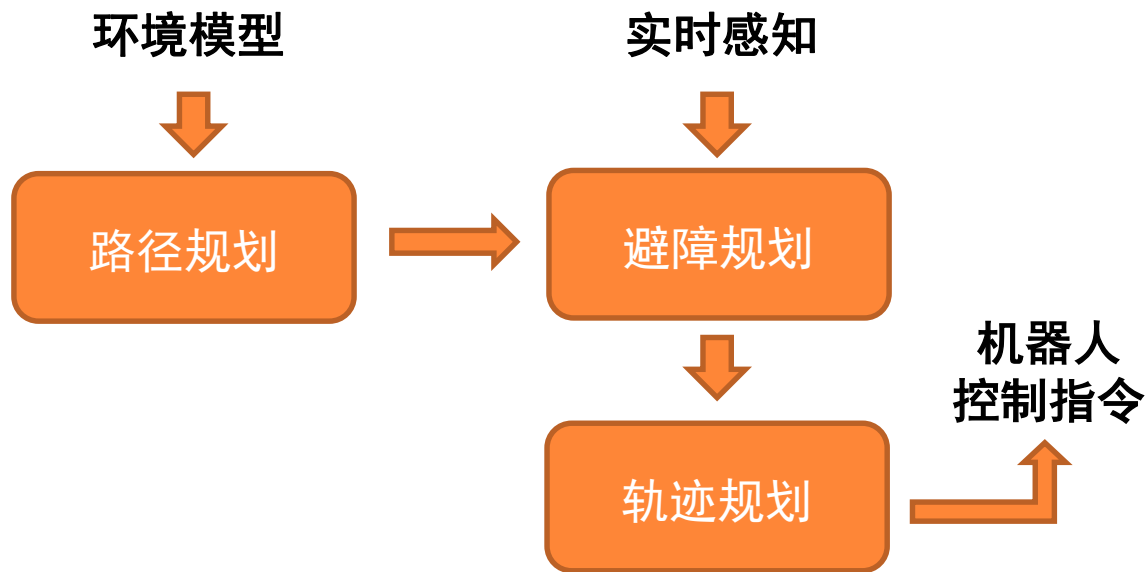
- **路径规划**：根据所给定的地图和目标位置，规划一条使机器人到达目标位置的路径（只考虑工作空间的几何约束，不考虑机器人的运动学模型和约束）
- **避障规划**：根据所得到的实时传感器测量信息，调整路径/轨迹以避免发生碰撞
- **轨迹生成**：根据机器人的运动学模型和约束，寻找适当的控制命令，将可行路径转化为可行轨迹。



路径规划、避障规划、轨迹规划三者关系



路径规划、避障规划、轨迹规划三者关系



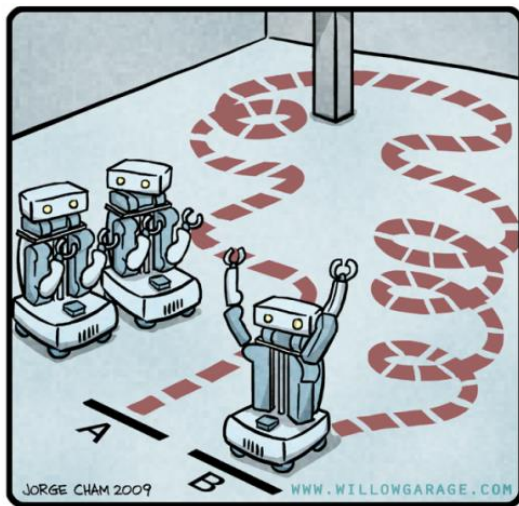
问题：实际应用中这三个模块都必须具备吗？



3.1 基本概念

路径规划

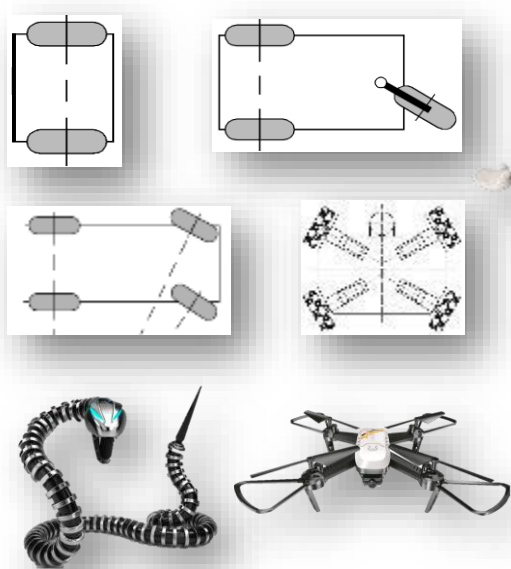
- 根据所给定的地图和目标位置，规划一条使机器人到达目标位置的路径



"HIS PATH-PLANNING MAY BE
SUB-OPTIMAL, BUT IT'S GOT FLAIR."

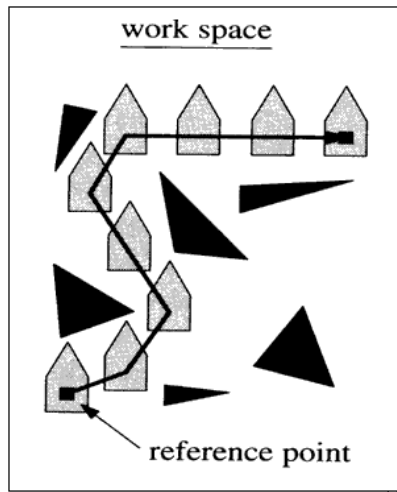
路径规划

- 简化：只考虑工作空间的几何约束，不考虑机器人的运动学模型和约束



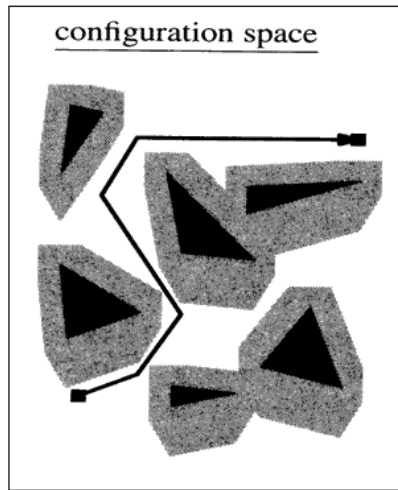
工作空间与位形空间(C-SPACE)

工作空间



工作空间：移动机器人上的参考点能达到的空间集合，机器人采用位置和姿态描述，并需考虑体积

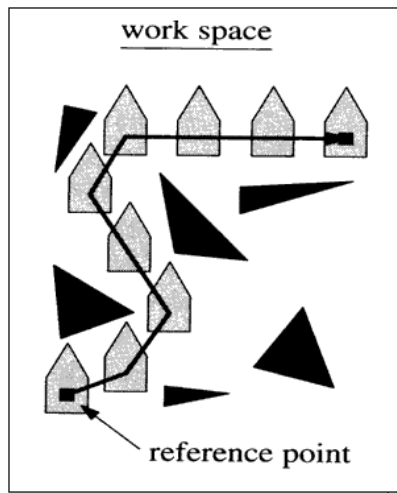
位形空间
(Configuration Space)



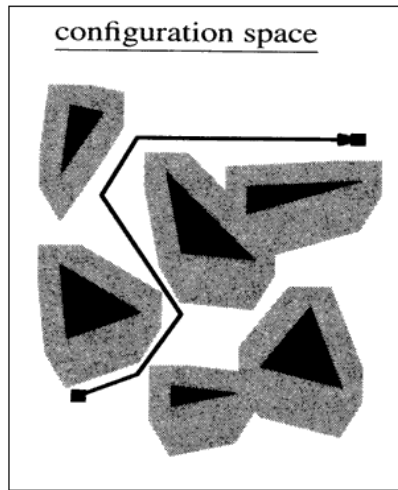
位形空间：机器人成为一个可移动点，不考虑姿态、体积和非完整运动学约束

工作空间与位形空间(C-SPACE)

工作空间



configuration space



位形空间
(Configuration Space)

障碍物按机器人半径进行膨胀

→ 机器人成为一个点

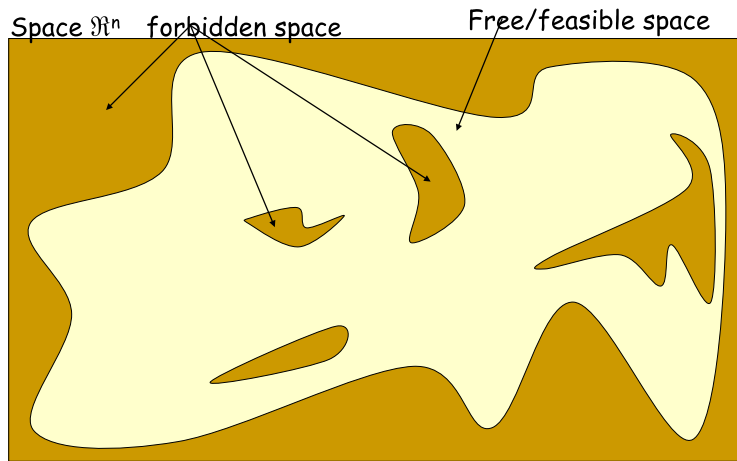
忽略非完整约束对姿态的限制

→ 机器人是完整的



位形空间

- 障碍物空间：不可行的位形集合
 - 在该空间中，机器人会与障碍物发生碰撞
- 自由空间：可行的位形集合
 - 在该空间中，机器人将无碰地安全移动

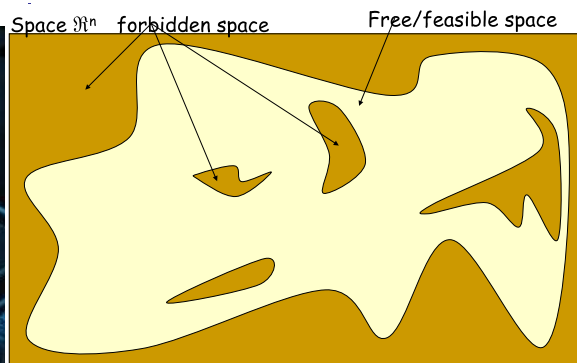
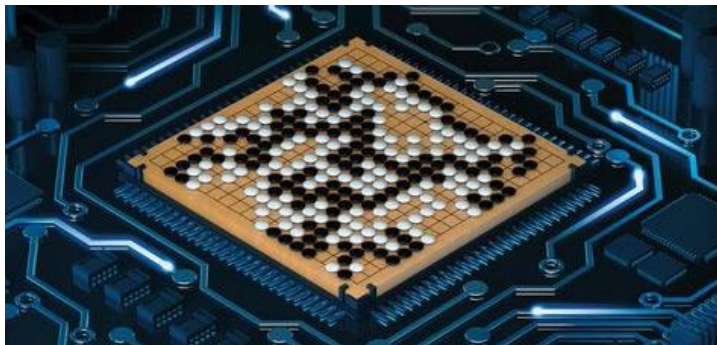


路径规划就是在自由位形空间中为机器人寻找一条路径，使其从初始位置运行到目标位置

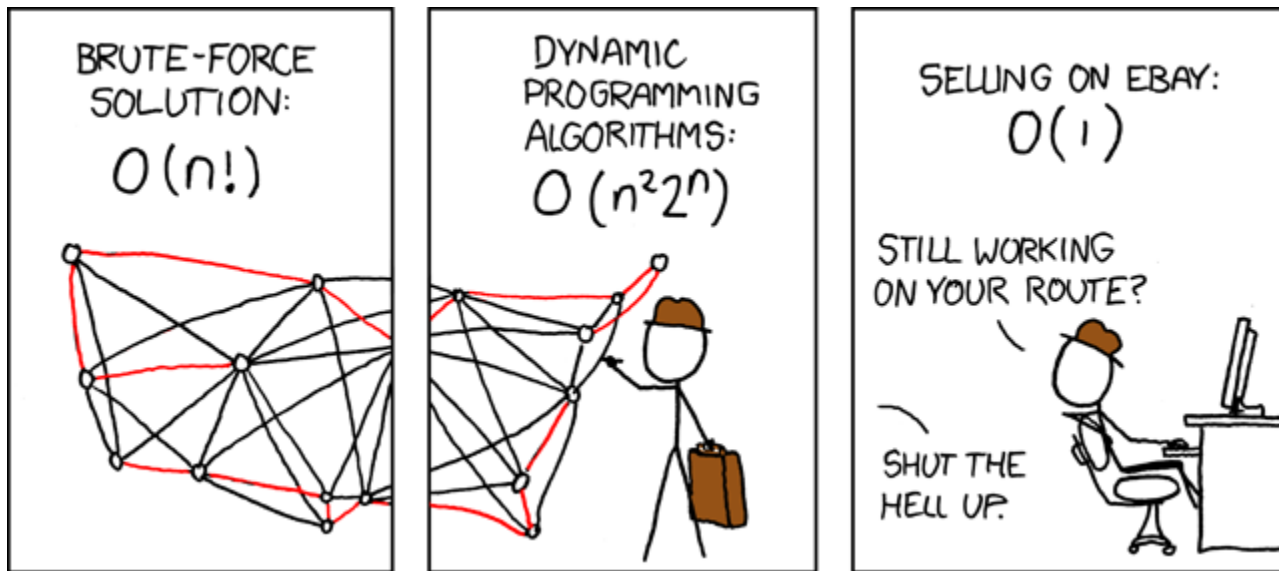


路径规划的完备性要求

- 完备性：当解存在时，能够在有限的时间内找到解
- 路径规划算法挑战：
 - 在连续空间内搜索，难以保证时间



一种常见的路径规划



拓扑连通图+最优路径搜索

路径规划的两个基本问题



拓扑连通图构建方法

- 基本思路：对空间作离散化
- **分辨率完备resolution completeness:**
解析性离散化，确保获得可行解
 - 行车图法：基于障碍物几何形状分解姿态空间
 - 单元分解法：区分空闲单元和被占单元
 - 势场法：根据障碍物和目标对空间各点施加虚拟力
- **概率完备Probabilistic completeness:**
基于概率进行随机采样离散化，使获得解的概率趋近于1
 - PRM (Probabilistic RoadMaps)
 - RRT (Rapid-Exploring Random Trees)



最优路径搜索方法

- 精确最优搜索法：深度优先法、宽度优先法
- 近似最优搜索法
 - 启发式搜索法：A*，D*
 - 准启发式搜索算法：退火、进化和蚁群优化等

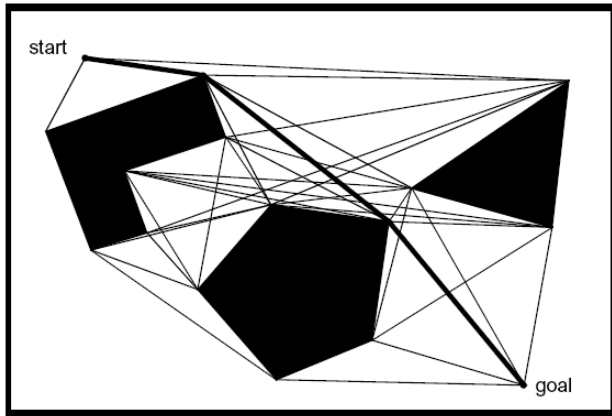




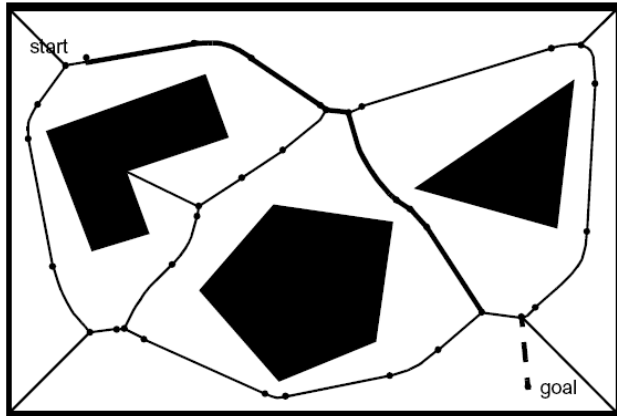
3.2 分辨率完备的连通图构建

1. 行车图法

- 基本思想：基于障碍物几何形状分解位形空间，将自由空间的连通性用一维曲线的网格表示，在加入起始点和目标点后，在该一维无向连通图中寻找一条无碰路径
- 构建行车图的典型方法：



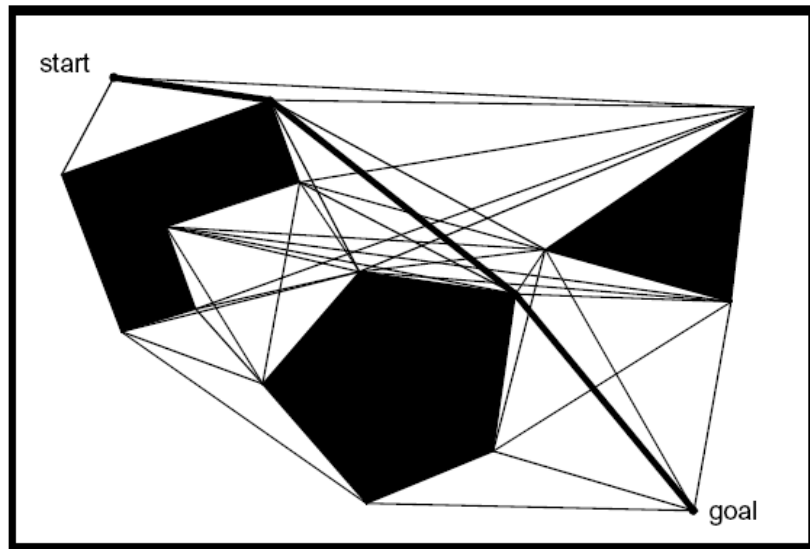
可视图 (Visibility graph)



Voronoi diagram

1.1 可视图法

- 可视图由所有**连接可见顶点对的边**组成
 - 可见指顶点之间无障碍物
 - 初始位置和目标位置也作为顶点



1.1 可视图法

○ 优点:

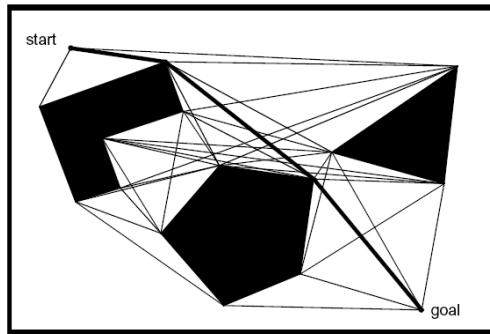
- 非常简单，特别是当环境地图用多边形描述物体时
- 可得到在路径长度上最优的解

○ 缺点:

- 所得路径过于靠近障碍物，不够安全。

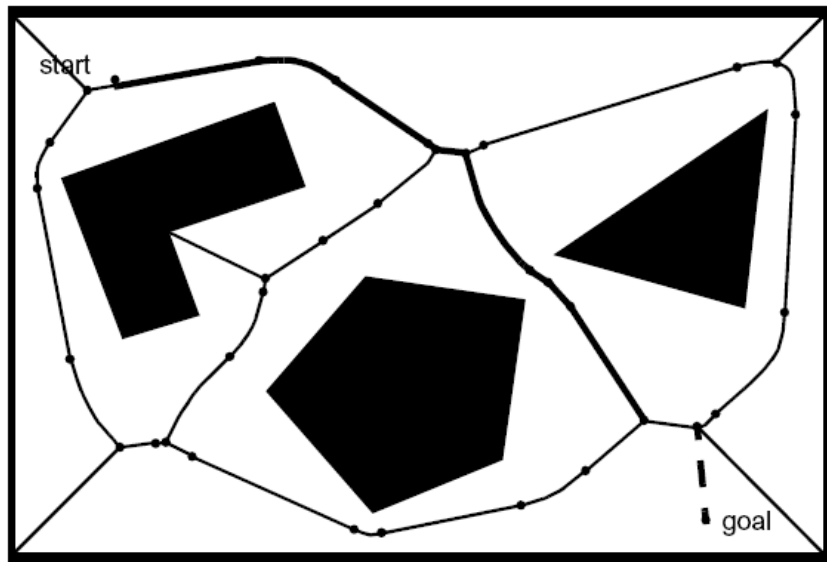
常用的解决方法:

- 以远大于机器人半径的尺寸膨胀障碍物，但容易造成可行路径的消失
- 在路径规划后修改所得路径，使其与障碍物保持一定的距离



1.2 Voronoi diagram

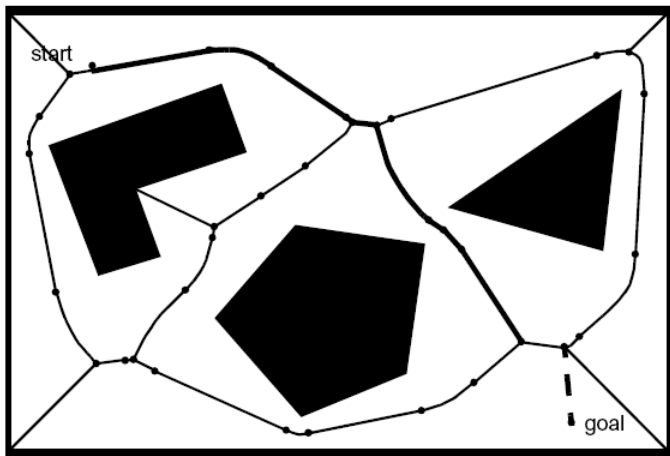
- 基本思想：取障碍物之间的中间点，以最大化机器人和障碍物之间的距离



1.2 Voronoi diagram

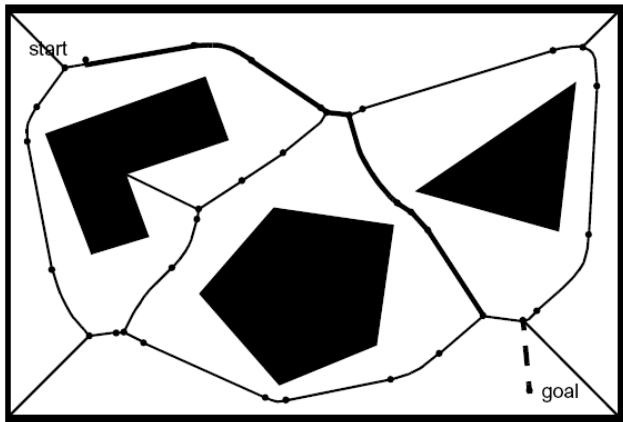
○ 构建方法：

- 对于自由空间中的每一点，计算它到最近障碍物的距离；
- 在垂直于二维空间平面的轴上用高度表示该点到障碍物的距离，类似于画直方图；
- 当某个点到两个或多个障碍物距离相等时，其距离点处出现尖峰，Voronoi diagram就由连接这些尖峰点的边组成。



1.2 Voronoi diagram

- 优点：安全性高
- 缺点：计算复杂、路径长度较可视图法长、不适用于短距离定位传感器



2. 单元分解法

○ 基本思想

- 首先，将位形空间中的自由空间分为若干的小区域，每一个区域作为一个单元，**以单元为顶点、以单元之间的相邻关系为边**构成一张连通图；
- 其次，在连通图中寻找包含初始姿态和目标姿态的单元，搜索连接初始单元和目标单元的路径；
- 最后，根据所得路径的单元序列生成单元内部的路径

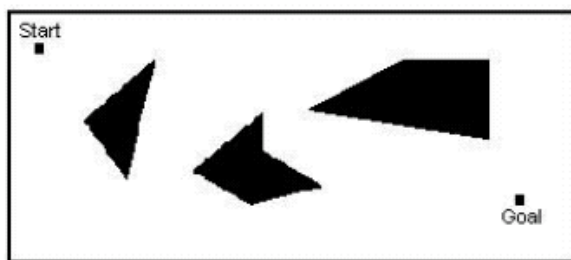
○ 主要方法

- 精确单元分解
- 近似单元分解

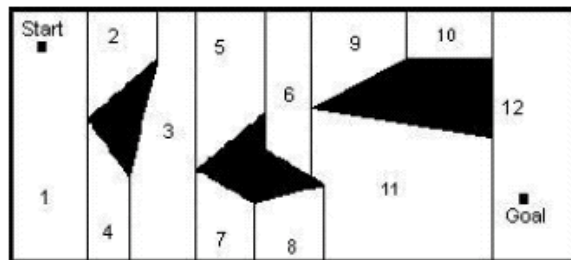


2.1 精确单元分解

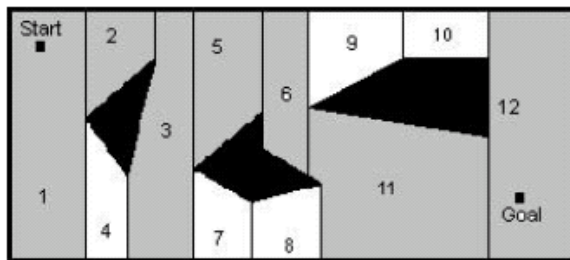
- 单元边界严格基于环境几何形状分解，所得单元完全空闲



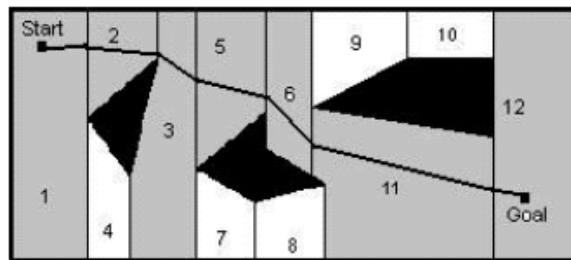
(a)



(b)



(c)



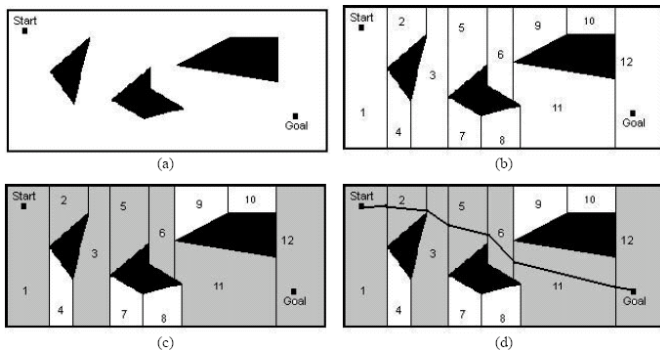
(d)

2.1 精确单元分解

○ 优点：

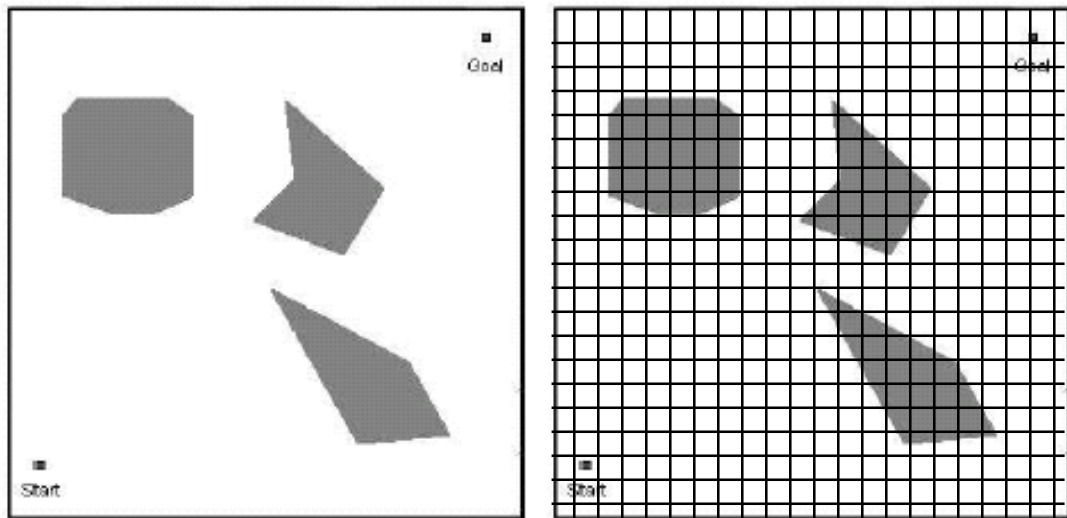
- 机器人不需要考虑它在每个空闲单元中的具体位置，只需要考虑如何从一个单元移动到相邻的空闲单元
- 单元数与环境大小无关

○ 缺点： 计算效率极大地依赖于环境中物体的复杂度



2.2 近似单元分解

- 栅格表示法，将环境分解成若干个大小相同的栅格



并不是每个单元都是完全被占或者完全空闲的，因此分解后的单元集合是对实际地图的一种近似



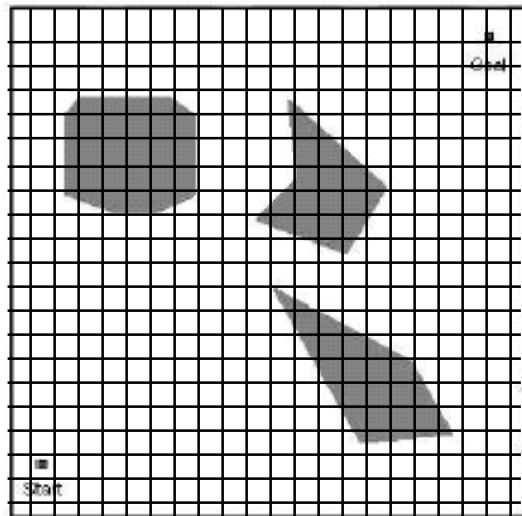
2.2 近似单元分解

○ 优点

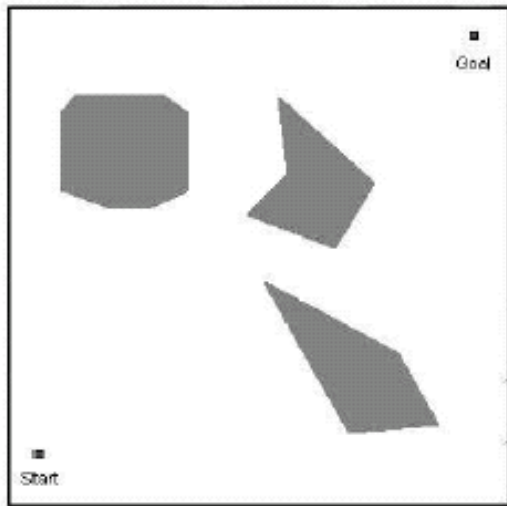
- 非常简单，与环境的疏密和物体形状的复杂度无关

○ 缺点：

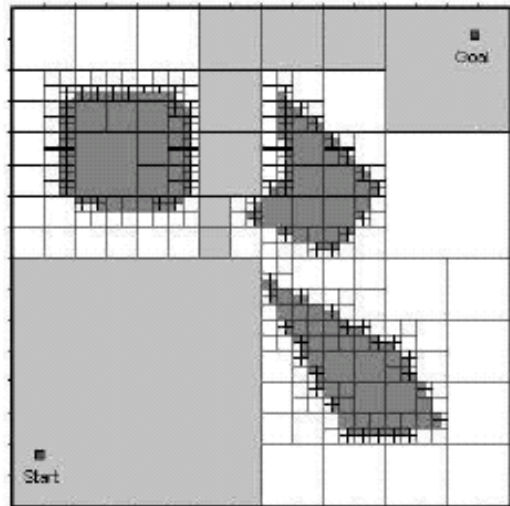
- 对存储空间有要求



可变大小的近似单元分解



(a)

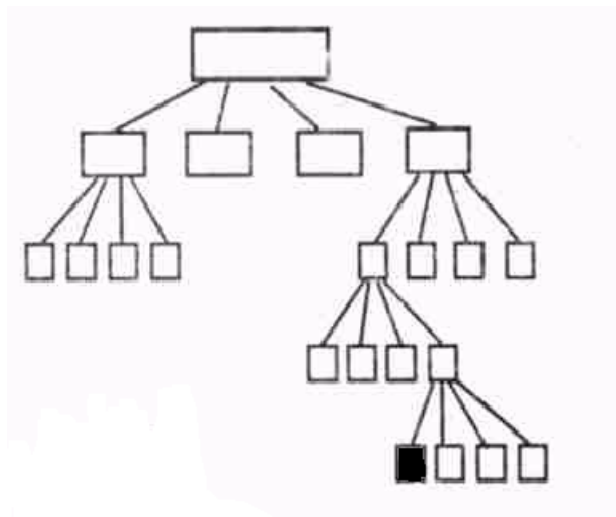


(b)

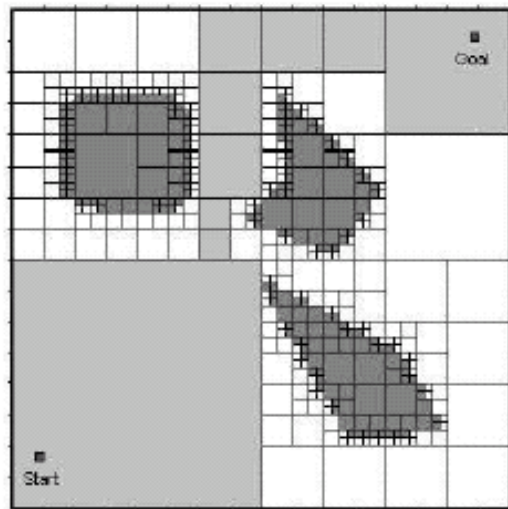
四叉树表示法：递归地把环境分为4个大小相等的子区域。直到每个区域中所包含的基本元素全为0或全为1。



可变大小的近似单元分解



(a)



(b)

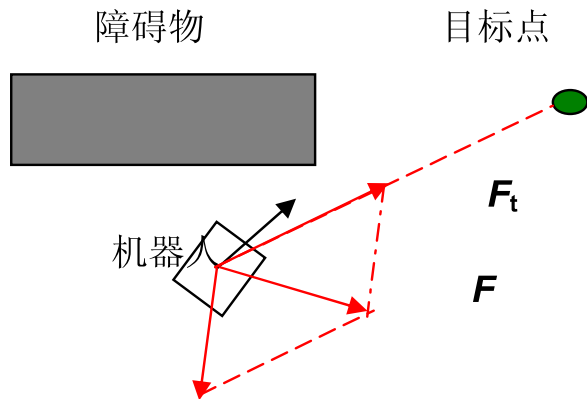
四叉树表示法：递归地把环境分为4个大小相等的子区域。直到每个区域中所包含的基本元素全为0或全为1。



3. 人工势场法

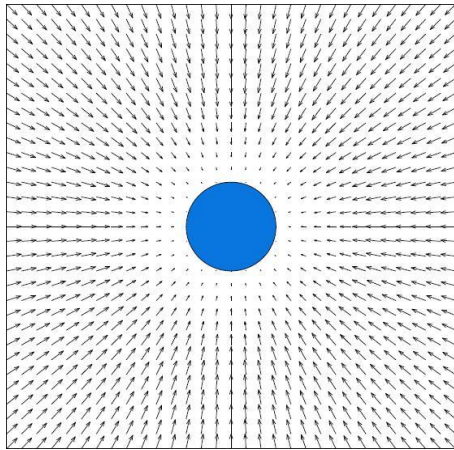
基本思想：

- 目标点对机器人产生吸引力，障碍物对机器人产生排斥力
- 所有力的合成构成机器人的控制律



3. 人工势场法

- 步骤1：构建人工势场(Artificial Potential Field)
 - 目标点：吸引势场



$$U_{att}(\mathbf{x}) = \begin{cases} K_a |\mathbf{x} - \mathbf{x}_d|^2 & |\mathbf{x} - \mathbf{x}_d| \leq d_a \\ K_a (2d_a |\mathbf{x} - \mathbf{x}_d| - d_a^2) & |\mathbf{x} - \mathbf{x}_d| > d_a \end{cases}$$

K_a 为系数

\mathbf{x} 为被评估点

\mathbf{x}_d 为目标点

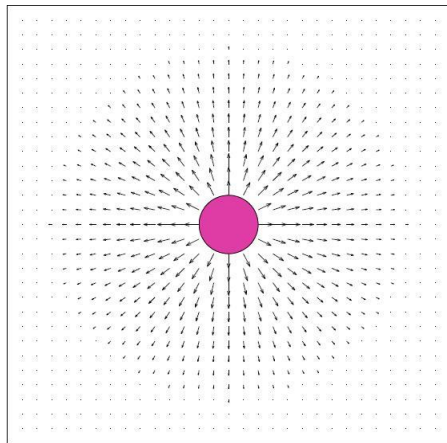
d_a 为距离阈值



人工势场法

步骤1：构建人工势场(Artificial Potential Field)

- 目标点：吸引势场
- 障碍物：推斥势场



$$U_{rep}(\mathbf{x}) = \begin{cases} \frac{1}{2} K_r \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 & \rho \leq \rho_0 \\ 0 & \rho > \rho_0 \end{cases}$$

r 被评估点和障碍物点之间的距离

r_0 预定义距离阈值



3. 人工势场法

○ 步骤2：根据人工势场计算力

● 对势场求偏导数

$$F_{att}(\mathbf{x}) = -\nabla U_{att}(\mathbf{x}) = \begin{cases} -2K_a(\mathbf{x} - \mathbf{x}_d) & |\mathbf{x} - \mathbf{x}_d| \leq d_a \\ -2K_a d_a \frac{\mathbf{x} - \mathbf{x}_d}{|\mathbf{x} - \mathbf{x}_d|} & |\mathbf{x} - \mathbf{x}_d| > d_a \end{cases}$$

$$F_{rep}(\mathbf{x}) = -\nabla U_{rep}(\mathbf{x}) = \begin{cases} K_r \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2} \frac{\partial \rho}{\partial \mathbf{x}} & \rho \leq \rho_0 \\ 0 & \rho > \rho_0 \end{cases}$$

$$\frac{\partial \rho}{\partial \mathbf{x}} = \left(\frac{\partial \rho}{\partial x} \quad \frac{\partial \rho}{\partial y} \right)^T = \frac{\mathbf{x} - \mathbf{x}_0}{\rho}$$

\mathbf{x}_0 为最近障碍物的坐标向量



3. 人工势场法

- 步骤3：计算合力，并进而由力计算得到控制律

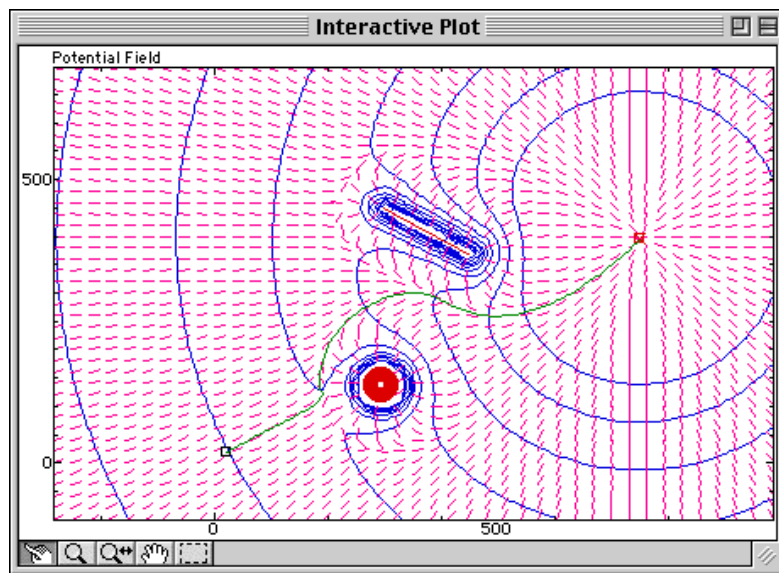
$$\begin{aligned} F(\mathbf{x}) &= -\nabla U(\mathbf{x}) \\ &= -\nabla U_{att}(\mathbf{x}) - \nabla U_{rep}(\mathbf{x}) \\ &= F_{att}(\mathbf{x}) + F_{rep}(\mathbf{x}) \end{aligned}$$

力的方向就是机器人运动方向，大小可以对应加速度控制



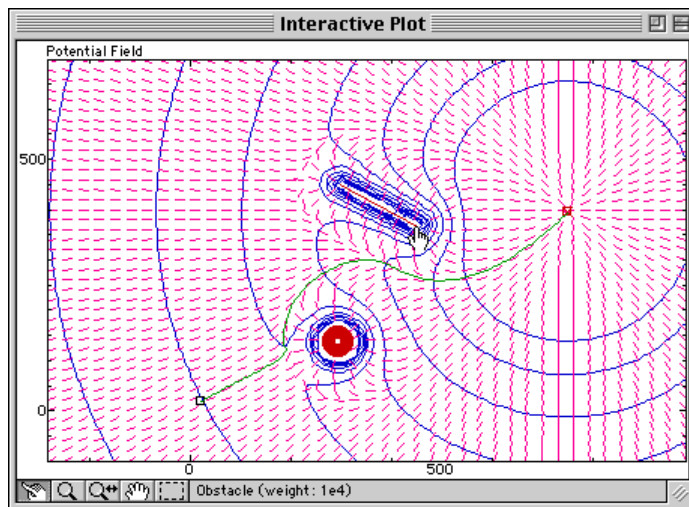
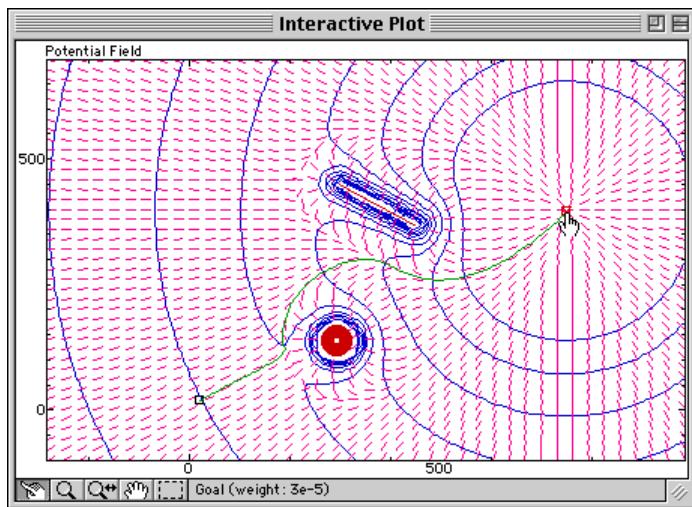
3. 人工势场法

- 机器人是受人工势场影响的一个点，沿着势场方向就可以避开障碍物达到目标点

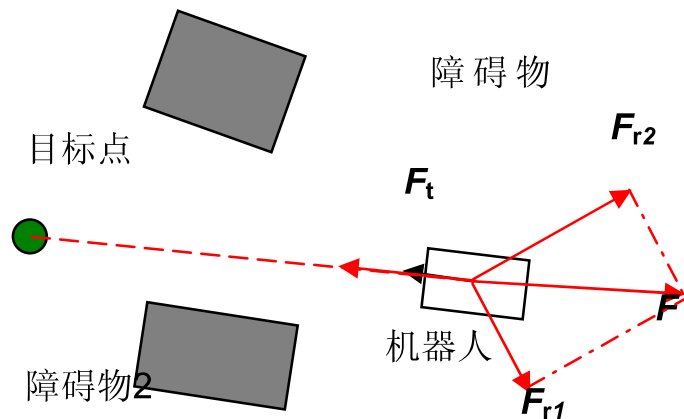


3. 人工势场法

- 不仅是一种路径规划方法，所构建的势场也构成了机器人的控制律，能够较好地适应目标的变化和环境中的动态障碍物，可以作为实时避障算法



3. 人工势场法



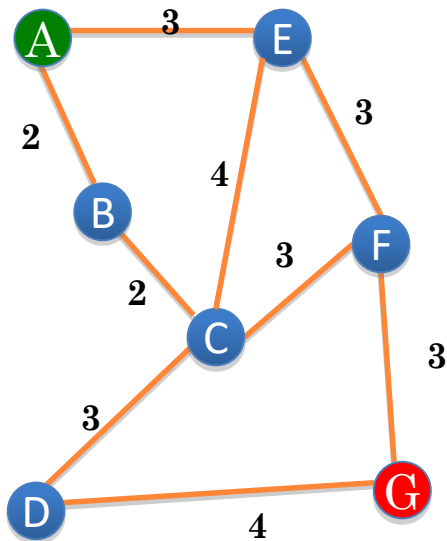
- 缺点：存在局部最小，容易产生振荡和死锁



3.3 路径搜索算法

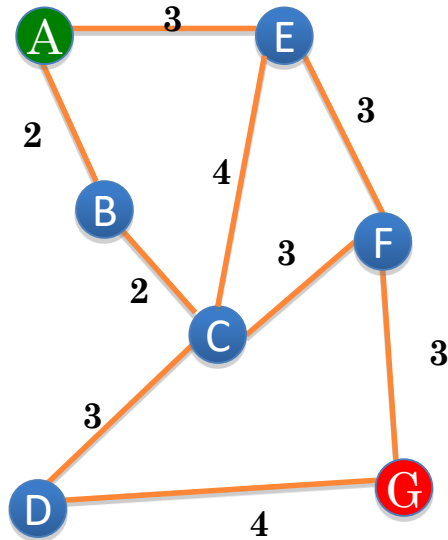
最优路径搜索算法

- 在构建形成的连通图中搜索最优路径

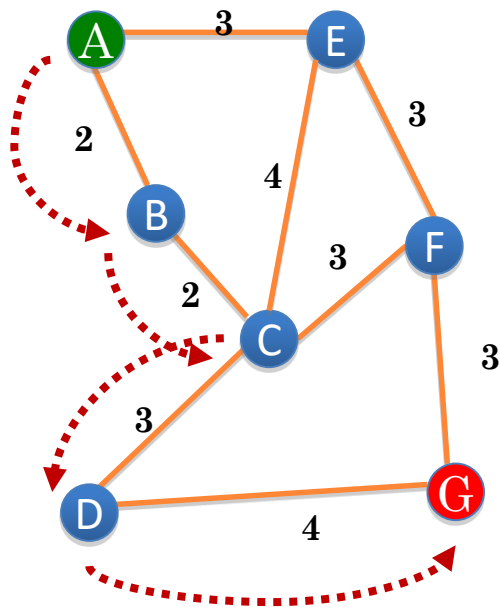


深度优先法

- 从起始节点开始按深度方式依次探索节点未被访问过的相邻节点
- 在一个相邻节点被访问后，优先访问该节点下一个未被访问相邻节点，直到扩展到图的最深层，即没有可访问的后继相邻节点，或者到达目标节点
- 返回上一层，探索该节点的其他未被访问相邻节点



深度优先法

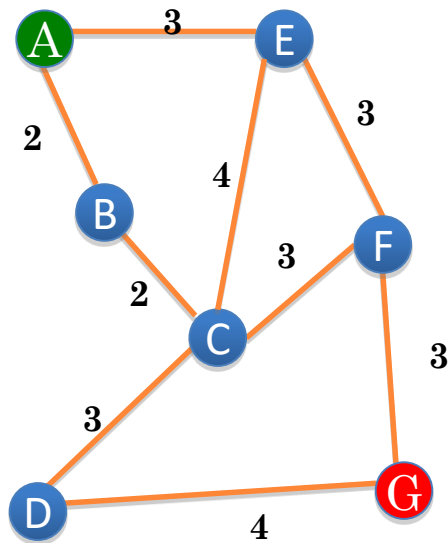


ABCDG 路径长11

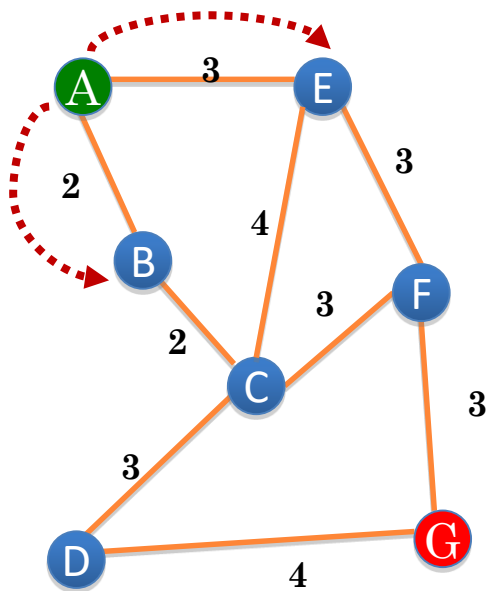


广度优先法

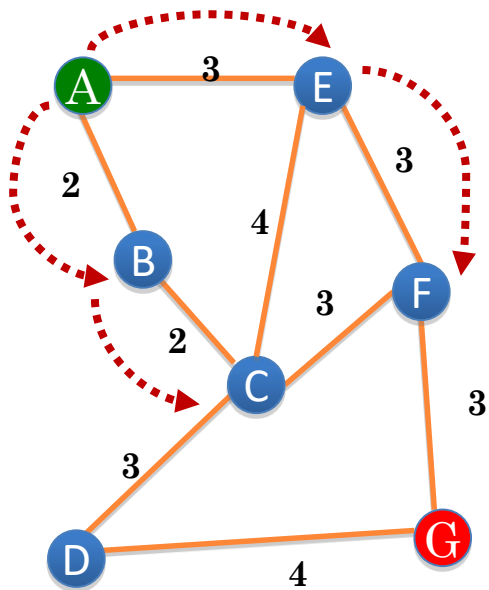
- 从起始节点开始访问该节点所有未曾访问的相邻节点
- 然后分别从这些相邻节点出发依次访问它们的相邻节点，访问优先级为先被访问的节点的相邻节点优先于后被访问的节点的相邻节点，直到所有被访问节点的相邻节点都被访问到



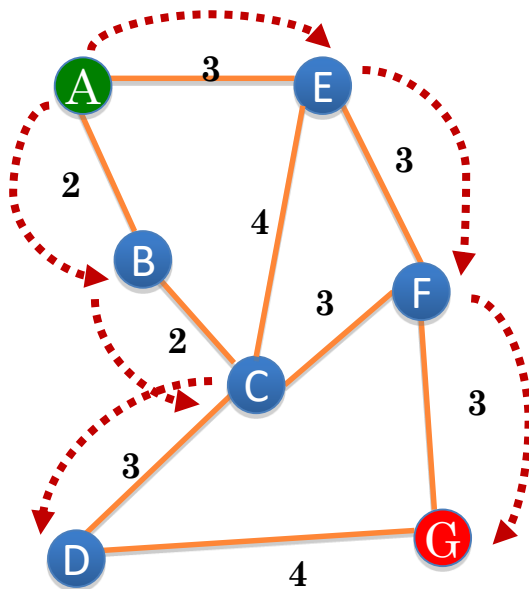
广度优先法



广度优先法



广度优先法

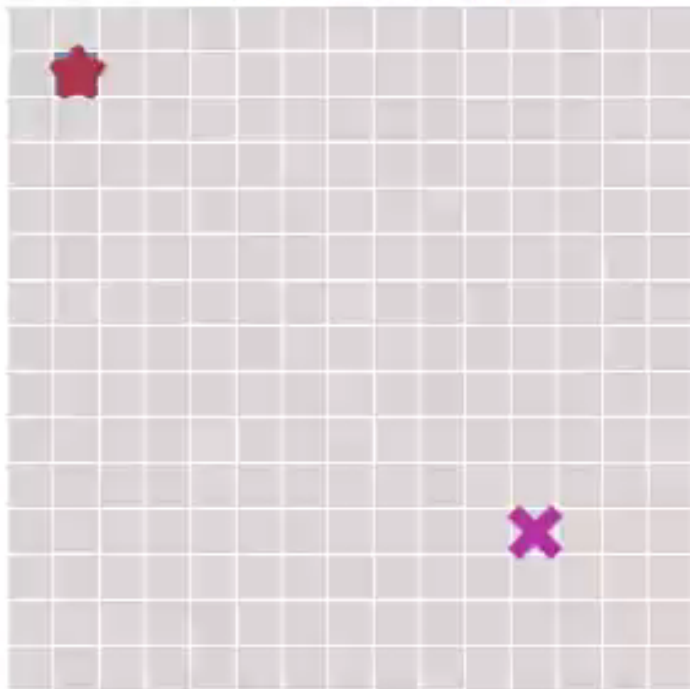


AEFG 路径长9

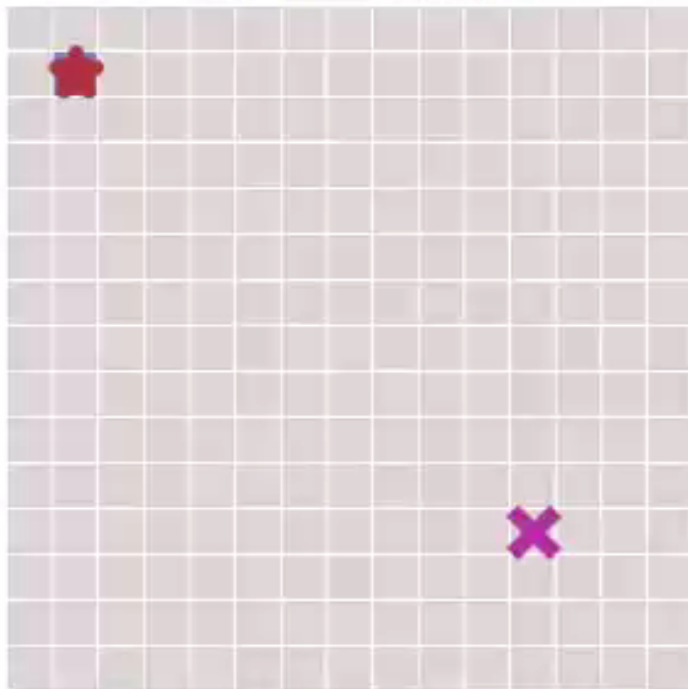


广度优先法与深度优先法比较

Breadth First Search



Depth First Search



广度优先法与深度优先法比较

○ 最优性：

- 广度优先和深度优先算法都不保证找到最优路径
- 当路径权重相等时，深度优先算法可以保证找到最优路径

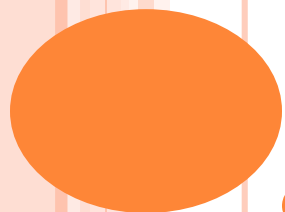
○ 完备性：

- 广度优先算法和深度优先算法都保证能够找到一条通过目标点的路径

○ 效率：

- 算法的复杂度相等
- 但在目标离起点近时，推荐使用广度优先，而在比较远时，推荐使用深度优先





END!

