# 侯立硕中期工作总结survey里问题定义

## survey目前如何定义问题，遇到的难点，论文解决了什么。

7里面每个agent都有唯一的优先级。

8 A Survey复杂环境机器人寻径，近十年的算法总结。出自transect science

1. Bringing Multi-agent Path Finding Closer to Reality

（1）Švancara, Jiří. Bringing Multi-agent Path Finding Closer to Reality[C]// Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

**（2）问题定义**

形式上，一个典型的MAPF实例可以写成一对（G,A），其中G=（V,E）是一个图，A是一组智能体，每个智能体∈A与起始位置∈V和期望目标位置∈V相关联。这意味着每个智能体也是一对=（,）。

时间是离散的，在每个时间步骤中，每个智能体都可以移动到相邻节点或保持在当前位置。用表示智能体的计划，然后(j)表示在时间步骤j中智能体存在的位置。MAPF问题的有效解决方案是的计划，以便满足以下约束：

1. 每个智能体的计划都是有效的路径即，如果(j)=v，(j+1)=u，则（v,u）∈E或(j)=(j+1)。

2. 没有两个智能体同时占用一个节点。即，对于所有时刻步骤j处的所有智能体对和，它认为(j)≠(j)。

3. 没有两个智能体同时占据一个边。即，对于所有时刻步骤j处的所有智能体对和，它认为(j)≠(j+1) V(j+1)≠(j)。

请注意，此定义允许智能体沿着完全占用的周期移动，只要它包含3个或更多节点。

除了有一个有效的解决方案外，通常还需要找到某个成本函数的最优解。最常用的两个函数是Sum of Costs (SOC)。

**（3）存在问题**

让路径规划更接近现实世界，经典的问题定义是假设智能体集是不变的，图中的距离是其次的，智能体之间没有冲突。

**（4）解决方法**

我们建议在问题规范中添加一组新的属性，使其更接近现实世界这些属性包括不同的距离、可以占用边或节点的智能体数，以及新智能体的动态外观。

1. The FastMap Algorithm for Shortest Path Computations

（1）Cohen L, Kumar T K S, Uras T, et al. The FastMap Algorithm for Shortest Path Computations[J]. 2017.

**（2）问题定义**

**（3）存在问题**

传统的FastMap适用于一般的无向图，许多传统的启发式方法，如Manhattan距离启发式，并没有很好的定义。

**（4）解决方法**

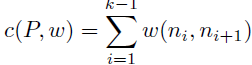
提出了一种新的预处理算法，将给定的边加权无向图的节点嵌入到欧氏空间中。这个空间中任意两个节点之间的欧几里得距离近似于给定图中它们之间最短路径的长度。之后，在运行时，任何两个节点之间的最短路径都可以通过A\*搜索来计算，使用欧几里德距离作为启发式。我们的预处理算法被称为FastMap，它受同名数据挖掘算法的启发，在近线性时间内运行。因此，FastMap比使用半定规划生成欧几里德嵌入的竞争方法快几个数量级，FastMap还生成可容许且一致的启发式算法，从而保证最短路径的生成。

1. Path Planning with CPD Heuristics

**（1）IJCAI-2019**

**（2）问题定义**

研究了边代价动态变化的加权图的路径规划问题。对于输入，我们需要一个带有节点N和边的图G=（N,E）。与图相关的是一个非负函数w:N\*N-> ，其中w（m,n）是边（m,n）∈E的权重。

路径规划问题的每个实例由开始节点s和目标节点t定义。我们的任务是在G中找到一条从s到t的路径，其中路径P被定义为一系列边,正是: 

我们认为在s到t的所有路径中，如果c（P,w）≤（,w），P是最优的，我们用[]表示空序列，用++表示序列连接。

**（3）存在问题**

当边缘成本是动态的，然而，快速路径规划算法不再保证最优性，并可能完全失败，由于无效的辅助数据。当发生这种情况时，辅助数据必须从零开始完全修复或重新计算。

**（4）解决方法**

压缩路径数据库（cpd）是一种在具有静态边缘代价的图中寻找最优路径的领先技术。在这项工作中，我们研究了cpd作为可接受的启发式函数，并将其应用于两种不同的情况：图受到动态变化成本的影响的问题，以及考虑时间有限的任何情况。

把CPD优化成能够解决动态边缘代价的算法。

1. Multi-Agent Path Finding on Ozobots

Ozobots是用于教程开发的小型移动机器人，文章是将用户路径规划的计划转换成Ozobots的命令序列。文章是讲Ozobots机器人的。

1. Implicitly Coordinated Multi-Agent Path Finding under Destination Uncertainty:Success Guarantees and Computational Complexity (Extended Abstract)

**（1）IJCAI-2019**

**（2）问题定义**

**（3）存在问题**

在多智能体路径发现（mapf）中，通常假设规划是集中执行的，并且智能体的目的地是公共知识。

**（4）解决方法**

我们将放弃这两个假设，并分析在哪些条件下可以保证代理使用隐式协调的计划在没有通信的情况下到达各自的目的地。

6、Improved Heuristics for Multi-Agent Path Finding with Conflict-Based Search

**（1）IJCAI-2019**

**（2）问题定义**

MAPF问题由一个无向未加权图G=(V,E)和一组k agent{…}指定，其中具有起始顶点∈V和目标顶点∈V。时间被离散成时间步。在连续的时间步之间，每个agent都可以移动到相邻的顶点或在其当前顶点处等待移动和等待操作都有单位成本，除非代理最终在其目标顶点等待，而目标顶点的成本为零。的路径是一系列的移动和等待动作，这些动作将从引导到。元组< >是顶点冲突，前提是和在时间步t处位于同一顶点v，并且元组<>是边冲突，前提是和在时间步t和t+1之间沿相反方向穿过同一边(u,v)。本文的目标是找到一组无冲突的路径，使所有的agent从起始点移动到目标点，同时使这些路径的代价和最小化。

**（3）存在问题**

基于冲突的搜索（cbs）及其增强是多智能体路径搜索中最强大的算法之一。最近的工作引入了一种可容许的启发式方法来指导cbs的高级搜索，在这项工作中，我们证明了这种启发式方法的局限性，因为它只基于基数冲突。

**（4）解决方法**

通过对代理之间的成对依赖关系进行推理，引入了两种新的可容许启发式算法。从经验上讲，使用两种新的启发式方法的cbs比使用最新的启发式方法的cbs显著提高了成功率，并将扩展节点和运行时的数量减少了50倍。

7、Priority Inheritance with Backtracking for Iterative Multi-agent Path Finding

(1) Okumura K , Machida M , Défago, Xavier, et al. Priority Inheritance with Backtracking for Iterative Multi-agent Path Finding[J]. 2019.

**（2）问题定义**

迭代多智能体寻路问题（迭代mapf）是多移动智能体寻路问题的一个推广，包括多智能体寻路（mapf）和多智能体取送（mapd），由于迭代mapf是一个抽象模型，我们不打算直接求解它，相反，必须根据目标域来体现任务创建。

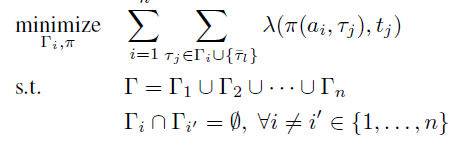
系统由一组agent（A={…}）和一个环境（如图G=（V,E）所示）组成，其中agent占据V中的节点并沿E中的边移动。考虑到实际情况，G必须是一个简单（既不是循环也不是多边）且强连接（每个节点都可以从每个其他节点到达）的有向图。这包括所有连接的简单无向图。

设(t)表示agent 在离散时间t所占用的节点，agent 的初始位置为(0)并作为输入给定。在每个时间步，选择一个节点(t+1)∈作为下一个时间步的位置。agent必须避免1）碰撞：；2）与其他agent相交：(t)≠(t+1)∨(t+1)≠(t)。我们不禁止旋转，所以(t+1)= 是可能的。

设是一组动态任务，随着时间的推移会添加新任务，即并非所有任务最初都已知。一个任务被定义为一个有限的目标集=，其中，可能依赖于上的一个偏序。当添加到设为时间步。agent没有分配任务时是空闲的。任务只能分配给自由agent。当被分配给时，开始访问中的目标。当所有的目标都实现了，就完成了，就自由了。

设现在是当在时间步t被分配到完成时的路径。当空闲时，它被分配一个带有路径的虚拟任务，其中索引*l*表示的唯一性。路径必须确保无碰撞和交叉，附加条件如下：服务时间定义为从开始（）到完成所经过的时间。注意通常不等于，因为并不总是立即赋值的。

我们说迭代mapf有一个解决方案，当且仅当所有任务的服务时间是有限的。迭代mapf是一个一般性的问题，它依赖于具体的实例，可以依赖于以下一般形式的目标函数。



其中是分配给的一组任务。迭代mapf的具体实例包括路径规划和任务分配，其目标函数在第5节中描述。

根据上下文，是先验的。例如，在mapf的文献中，被提供为{={}}，其中是的目标节点。开始节点由(0)定义。在mapf的基本定义中，当所有agent在同一时间步内达到其目标时，终止就实现了。为了满足这个需求，当到达的离开它时，一个新的任务被添加到中。

MAPD是迭代MAPF的一个实例，其中每个任务都是由两个节点（拾取和传递）组成的元组，而不是一个集合。

**（3）存在问题**

MAPF这个问题的实际应用中，例如在自动化仓库中导航，必须迭代地解决MAPF。

**（4）解决方法**

本文提出了一种迭代映射函数的新方法，称为带回溯的优先级继承（PIBT）。PIBT在每个时间步给每个agent一个唯一的优先级，这样所有的移动都被优先化。优先级继承的目的是在一个小的时间窗内有效地处理路径调整中的优先级反转，它可以迭代地应用，并且回溯协议可以防止agent被卡住。我们证明，当环境是一个图，使得所有相邻节点对都属于一个长度为3或3以上的简单循环（例如，双连接）时，无论其数目如何，所有代理都保证在有限时间内到达它们的目的地。我们的pibt实现可以完全分散，而无需全球通信。在不同场景下的实验结果证实，pibt既可以在有多个代理的大型环境中查找路径，也可以在自动化仓库中传输包。

8、Path Planning Algorithm in Complex Environment: A Survey

对于复杂环境下的自主虚拟机器人导航来说，路径发现算法是一个极具挑战性的问题。一个可靠的导航系统必须能够识别虚拟机器人的当前位置，避免任何碰撞，并确定物体的平滑路径轨迹。目前，为了实现自主虚拟机器人的避障路径规划算法中的若干问题，需要产生具有令人印象深刻的碰撞方案的系统和高效的路径发现算法，使得研究者进行各种实验来改进和修改现有算法。为了解决复杂环境下自主虚拟机器人的导航问题，提出了近10年来的一系列路径规划算法。我们相信，本文所研究的各种算法将为虚拟环境、碰撞检测和机器人领域的研究人员提供一些基本背景，以及在这种复杂环境下自主虚拟机器人导航过程如何工作的问题和挑战。

介绍的算法：*Path planning using corridor maps，The bug algorithm，Follow the Gap Method (FGM)，New Hybrid Navigation Algorithm (NHNA)，Intelligent Bug Algorithm (IBA)，Intelligent Follow the Gap Method。*

9、E-Bug: New Bug Path-planning algorithm for autonomous

robot in unknown environment

（1）Meddah F, Dib L. P\*: A New Path Planning Algorithm for Autonomous Robot in an Unknown Environment[C]// International Conference on Intelligent Information Processing. 2015.

（2）文章对于E-Bug和以往的Bug算法做了比较。重点写了各种算法的不足。E-Bug用Euclidian距离选择最短路径。

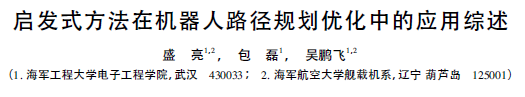
10、A survey on coverage path planning for robotics

（1）Galceran E , Carreras M . A survey on coverage path planning for robotics[J]. Robotics and Autonomous Systems, 2013, 61(12):1258-1276.

（2）本文是CPP问题的Survey.覆盖路径规划（cpp）是指在避开障碍物的情况下，确定一条经过感兴趣区域或体积的所有点的路径。本文回顾了国内外最成功的cpp方法，重点介绍了近十年来所取得的成就。讨论了所述cpp方法的现场应用。

11、启发式方法在机器人路径规划优化中的应用综述

（1）盛亮, 包磊, 吴鹏飞. 启发式方法在机器人路径规划优化中的应用综述[J]. 电光与控制, 2018(9):58-64.

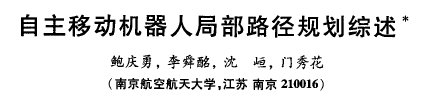


（2）



12、自主移动机器人局部路径规划综述

（1）鲍庆勇, 李舜酩, 沈峘, et al. 自主移动机器人局部路径规划综述[J]. 传感器与微系统, 2009, 28(9):1-4.



（2）主要讲局部路径规划：

局部路径规划传统方法：模拟退火算法，人工势场法，模糊逻辑算法

局部路径规划智能仿生算法：人工神经网络算法，遗传算法，蚁群优化算法，粒子群算法

局部路径规划启发式搜索方法：

局部路径规划基于滚动窗口的算法

局部路径规划基于行为的路径规划算法

局部路径规划基于再励学习的路径规划算法

## 二、A\*代码实现

（袁玥说找到方法可以在平台上运行Python了，因此Python的A\*算法也在调试）

1、C#的已经找到代码，正在调试。

2、Python的已经找到代码，正在调试

## 三、LATEX安装使用

1、安装了CTEX，对刚接触的比较友善。

2、推荐安装的是TEX Live,准备再安装一个TEX Live。

2专业名词

**（1）引擎**

引擎是将文档的内容进行排版的真正主体，包含以下几种：

TeX: 最为原始的排版引擎，生成的是.dvi文件，现在很少用到。

pdfTeX: 在TeX的基础上将生成的.dvi文件转化成.pdf文件，是不少发行版的默认引擎。

XeTeX(推荐): 是使用了Unicode的TeX排版引擎，对于中文的支持前所未有的好。除此以外还支持了不少现代字体技术(OpenType等)，排版效果有一定的提升。

LuaTeX: 最新的排版引擎，但是默认使用Lua，对于LaTeX的支持现在并不完善。

（2）**宏集**

宏集类似于编译器，可以将较为接近自然语言的内容编译成排版引擎可以“理解”的内容。以下是一些比较常见的宏集。

plain TeX: 是最古老的TeX宏集, 提供了一些基本的命令，语法和现在流行的LaTeX已经大有不同。

LaTeX: Leslie Lamport开发的宏包，更加直观简单。

AMSTeX: 美国数学会提供的一个TeX宏集，针对数学符号和公式进行了优化

（3）**发行版**

发行版是一个TeX引擎、宏集、文档模板的集合，是使用LaTeX不得不安装的一个组件。现在常见的发行版有：

TeX Live(推荐): 是由国际TeX用户组织开发的TeX系统，是更新最快，内容最全的发行版。支持Windows, Unix, Linux。

MacTeX: 同为TUG开发的TeX系统，针对Mac OS进行开发，安装简单、功能齐全。

CTeX: 中科院吴凌云研究员基于MiKTeX开发的TeX系统，方便了中文输入。但是从2012年后已经停止开发不再更新，不推荐使用。