



## 多智能体路径规划研究进展

刘庆周, 吴 锋

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

**摘 要:** 多智能体路径规划是一类寻找多个智能体从起始位置到目标位置且无冲突的最优路径集合的问题, 针对该问题的研究在物流、军事和安防等领域有着大量的应用场景。对国内外关于多智能体路径规划问题的研究进展进行系统整理和分类, 按照结果最优性的不同, 多智能体路径规划算法被分为最优算法和近似算法 2 类。最优的多智能体路径规划算法主要分为基于 A\* 搜索、基于代价增长树、基于冲突搜索和基于规约的 4 种算法。近似的多智能体路径规划算法主要分为无边界次优的算法和有边界次优的算法 2 类。基于上述分类, 分析各种算法的特点, 介绍近年来具有代表性的研究成果, 并对多智能体路径规划问题未来的研究方向进行展望。

**关键词:** 多智能体路径规划; 人工智能; 搜索; 最优路径集合; 多机器人

开放科学(资源服务)标志码(OSID):



中文引用格式: 刘庆周, 吴锋. 多智能体路径规划研究进展[J]. 计算机工程, 2020, 46(4): 1-10.

英文引用格式: LIU Qingzhou, WU Feng. Research progress of multi-agent path planning[J]. Computer Engineering, 2020, 46(4): 1-10.

## Research Progress of Multi-Agent Path Planning

LIU Qingzhou, WU Feng

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

**【Abstract】** Multi-Agent Path Planning (MAPP) is the problem of finding an optimal path set for multiple agents from their starting position to the target position without any collisions. Research on this problem has abundant application scenarios in the fields of logistics, military and security. This paper systematically sorts out and classifies the recent research progress in MAPP both at home and abroad. According to the differences in the optimality of results, the MAPP algorithms are classified into optimal group and approximate group. The optimal MAPP algorithms are mainly divided into four categories, which are based on A\* search, cost growth tree, conflict based search and protocol respectively. The approximate MAPP algorithms are mainly divided into two categories; the unbounded suboptimal algorithm, and the bounded suboptimal algorithm. Based on the classification mentioned above, this paper analyzes the characteristics of each algorithm, introduces the representative researches in recent years and discusses the future research directions of MAPP problem.

**【Key words】** Multi-Agent Path Planning (MAPP); artificial intelligence; search; optimal path set; multi-robot

DOI: 10.19678/j.issn.1000-3428.0056738

### 0 概述

多智能体路径规划 (Multi-Agent Path Planning, MAPP) 问题是一类寻找多个智能体从起始位置到目标位置且无冲突的最优路径集合的问题, 其在机场拖航<sup>[1]</sup>、物流仓储<sup>[2]</sup>、交通控制<sup>[3]</sup>、机器人<sup>[4]</sup>、电子游戏<sup>[5]</sup>等领域都有广泛的应用。近年来, 研究人员在多智能体路径规划问题的研究中取得了众多突破性进展, 但国内对这些工作进行整理形成的综述较少, 最近的相关综述<sup>[6]</sup>只给出该问题的形式化定义和相

关算法的简单分类, 没有涉及算法的具体原理和最新进展。

在机器人领域, 存在一类被称作多机器人路径规划的问题。多机器人路径规划与多智能体路径规划 2 类问题存在许多相似之处, 例如, 两者都关注如何求解问题、提高求解效率与结果质量等。但是, 多机器人路径规划同时关注机器人的视觉、定位、通信、运动控制等实际约束和设定, 而多智能体路径规划则将这些设定进行抽象化, 其将研究的重点集中在问题的求解方式、求解效率和求解质量上。

基金项目: 国家自然科学基金青年基金“基于决策理论的半自主智能体决策规划模型和算法研究”(61603368)。

作者简介: 刘庆周 (1994—), 男, 硕士研究生, 主研方向为多智能体路径规划; 吴 锋, 副教授。

收稿日期: 2019-11-28 修回日期: 2020-02-05 E-mail: wufeng02@ustc.edu.cn

本文研究多智能体路径规划问题。首先给出多智能体路径规划的问题定义和问题属性,概述主流的4类最优多智能体路径规划算法,解析算法的基本原理,然后分析2类近似多智能体路径规划算法的优缺点。在此基础上,总结多智能体路径规划问题的现有求解算法并对未来的研究方向进行展望。

## 1 多智能体路径规划

### 1.1 问题定义

一个经典的多智能体路径规划问题可以被定义为一个四元组 $\langle G, k, S, T \rangle$ 。其中, $G = (V, E)$ 是一个无向图,无向图中的节点 $v \in V$ 是智能体可以占据的位置,边 $e = (v_i, v_j) \in E$ 是 $v_i$ 和 $v_j$ 之间的连线,表示智能体可以在 $v_i$ 和 $v_j$ 之间移动。 $k$ 代表问题中的智能体数量,即智能体 $\{a_1, a_2, \dots, a_k\}$ 。每个智能体 $a_i$ 都有独一无二的起始位置 $s_i \in S \in V$ 和目标位置 $g_i \in T \in V$ 。 $S$ 是所有智能体初始位置的集合, $T$ 是所有智能体目标位置的集合。

在多智能体路径规划问题中,时间被离散化为时间步的形式。在每个时间步中,每个智能体可以采取一次行动。一般来说,有等待和移动2种行动。在多智能体路径规划问题中,存在智能体之间的冲突,且冲突主要有2种,即图1所示的碰撞冲突和图2所示的交换冲突。

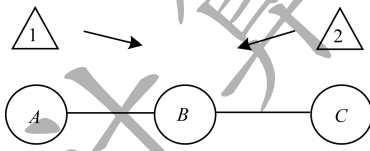


图1 碰撞冲突  
Fig.1 Crash collision

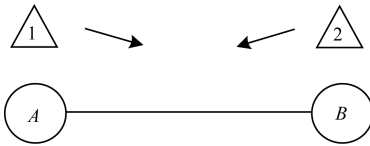


图2 交换冲突  
Fig.2 Swap collision

智能体 $a_i$ 从 $s_i$ 移动到 $g_i$ 的行动序列构成一条路径 $p_i$ 。多智能体路径规划问题的一个可行解是 $k$ 个智能体的 $k$ 条路径的集合 $P = \{p_1, p_2, \dots, p_k\}$ ,其

中,智能体 $a_i$ 对应于路径 $p_i$ ,路径集合中的任意2条路径 $p_i$ 和 $p_j$ 之间不存在冲突。

多智能体路径规划问题通常都需要最小化某个全局累积代价函数,常用的代价函数有如下4种:

$$\max_{1 \leq i \leq k} t_i \quad (1)$$

$$\max_{1 \leq i \leq k} l(p_i) \quad (2)$$

$$\sum_{i=1}^k t_i \quad (3)$$

$$\sum_{i=1}^k l(p_i) \quad (4)$$

代价函数式(1)表示最晚到达目标位置的智能体所花费的时间,代价函数式(2)表示路径集合中长度最长的路径长度,代价函数式(3)表示所有智能体分别从各自起始位置到达各自目标位置所花费时间之和,代价函数式(4)表示路径集合中所有路径长度之和。

### 1.2 问题属性

多智能体路径规划问题的状态空间随着问题中智能体的增多而指数增长,该问题也被证明是NP-hard问题<sup>[7]</sup>。因此,最优的多智能体路径规划算法只有在智能体相对较少的情况下才有较好的实际应用价值。尽管如此,最优的多智能体路径规划算法仍然有很大的研究价值。适当牺牲多智能体路径规划算法的最优性可以大幅提高算法的执行效率,这也是有边界的次优多智能体路径规划算法的核心思想。除此之外,对最优多智能体路径规划算法的研究可以更深刻地揭示多智能体路径规划问题的特点。

在多智能体路径规划问题的研究中,有2个非常重要的性质,即完整性和最优性。完整性指当问题存在解时,算法能够找到可行解;当问题不存在解时,算法能够终止。最优性指当问题有解时,算法返回的一定是最优解。一般来说,最优的多智能体路径规划算法需要同时满足完整性和最优性,而近似算法没有这种要求,一部分近似算法具有完整性,另一部分则不具备完整性。

## 2 最优多智能体路径规划算法

如表1所示,最优的多智能体路径规划算法可以分为A\*搜索的扩展、代价增长树搜索、基于冲突的搜索和基于规约的方法4类。

表1 最优多智能体路径规划算法对比

Table 1 Comparison of optimal MAPP algorithms

算法分类	优点	缺点	适用问题规模	适用问题特点
A*搜索算法	实现简单,在智能体密集问题中表现良好	时间代价和空间代价高,速度最慢	中小规模(2个~30个智能体)	在智能体密度较高时效果较好,密度越高效果越好,对密度敏感性较低
代价增长树搜索算法	实现比较简单,分为2层搜索,速度较快	速度偏慢,高层次搜索冗余度高	中等规模(2个~60个智能体)	在智能体密度较低时效果好,较高时计算代价高
基于冲突搜索的算法	速度相对较快,求解问题规模较大	实现难度略高	中等规模(2个~60个智能体)	适用于智能体稀疏问题,在加入合并操作后适用于任意密度问题
规约算法	速度一般很快,只要完成证明就可以找到高效的求解器	规约证明需要极强的数理功底	中等规模(2个~60个智能体)	由规约后的模型特点决定

## 2.1 A\* 搜索的扩展

### 2.1.1 多智能体路径规划中的 A\* 搜索

A\* 搜索是一种经典的搜索算法, 也同样适用于多智能体路径规划问题。A\* 搜索在多智能体路径规划中的状态空间一般被称为  $k$ -agent 状态空间, 其状态数等于将  $k$  个智能体分别放置在  $|V|$  个不同的节点上的状态数, 一种放置方法对应于一种状态。A\* 搜索的初始状态节点表示所有智能体都在各自的初始位置, 目标状态节点表示所有智能体都到达各自的目标位置。A\* 搜索中状态的后继表示所有智能体各自采取一组无冲突的行动从当前状态转移到新的状态。

对 A\* 搜索启发式函数的研究在多智能体路径规划中非常重要。最简单的全局启发式函数是所有智能体各自的启发式函数之和, 这在格点图上表现为曼哈顿距离之和, 而在欧几里得图上表现为欧几里得距离之和<sup>[8]</sup>。在很多的智能体路径规划研究中, 采用了更为复杂的全局启发式函数, 如 SIC<sup>[9]</sup> 和基于模式-数据库的启发式函数<sup>[10-11]</sup>。然而, 采用过于复杂的启发式函数不一定具有积极意义, 因为提高启发式函数的计算复杂度可能会降低算法的执行效率。

### 2.1.2 A\* 搜索的缺陷和改进

在多智能体路径规划问题的求解中, A\* 搜索主要有两大缺陷, 这些缺陷也使得 A\* 的求解能力严重受到问题中智能体数量的制约。第 1 个缺陷是搜索状态空间的指数增长, A\* 搜索的状态空间随着问题中智能体的增多而指数增长, 这使得 A\* 中的 open-list 在智能体过多的情况下会发生溢出; 第 2 个缺陷是搜索分支因子的指数增长, 由于每个智能体在每个时间步能够采取  $b$  ( $b > 1$ ) 种行动, 扩展到多个智能体时, A\* 搜索的分支因子也会随着智能体的增多而指数增长, 即  $b^k$ 。上述缺陷导致 A\* 搜索无法求解较多智能体的路径规划问题。

目前, 对 A\* 搜索的缺陷进行改进主要通过 4 种方法, 即 OD (Operator Decomposition)<sup>[9]</sup>、EPEA\* (Enhanced Partial Expansion)<sup>[10-11]</sup>、ID (Independence Detection)<sup>[9]</sup> 和 M\*<sup>[12]</sup>, 4 种方法的对比见表 2。

表 2 A\* 算法的主要改进方法

Table 2 Main improvement methods of A\* algorithm

改进方法	状态空间缺陷	分支因子缺陷
OD	未改进	通过每次只扩展单个智能体并引入中间节点来改进缺陷
EPEA*	未改进	只扩展最优节点, 当前节点修改启发式函数值并重新加入 open-list
ID	通过计算关联性, 将问题分解为子问题进行求解	未改进
M*	处于最优状态的智能体的空间维度为 1	因为部分智能体处于最优状态, 整体减小了分支因子

OD 改进的是 A\* 搜索中分支因子指数增长的缺陷。当 A\* 搜索扩展初始状态  $(s_1, s_2, \dots, s_k)$  时, 只考虑第 1 个智能体的下一步行动, 这样会生成当前

状态的一系列后继状态, 即一共  $b$  个后继状态。在这些后继状态中, 只有第一个智能体执行行动, 其余的  $k-1$  个智能体仍然保持  $t=0$  时刻的位置。将生成的这些后继状态节点加入到 open-list 中, 当扩展这些状态节点时, 只考虑第 2 个智能体的行动, 并生成新的一系列后继状态。新生成的后继状态代表第 1 个和第 2 个智能体可能处于的所有位置, 其他智能体仍然处于  $t=0$  时刻的位置。按照这样的规则继续进行搜索, 直至搜索树上的第  $k$  层状态节点能够表示所有智能体在  $t=1$  时刻的所有可能位置分布。第  $k$  层状态节点表示同一时刻所有智能体的位置分布, 这类状态节点被称为满状态节点, 其他状态节点被称为中间状态节点。搜索在到达满状态节点  $(g_1, g_{t+1}, \dots, g_k)$  时终止。OD 大幅减小了 A\* 搜索的分支因子。原始 A\* 的分支因子是  $b^k$ , 而 ODA\* 的分支因子是  $b$ 。相应地, ODA\* 也引入了大量的中间状态节点, 并且使得 A\* 搜索的深度加深了  $k$  倍。然而, 在多智能体路径规划中, 这些中间状态节点一般会因为其启发式函数值过大而被剪枝。因此, OD 的引入起到了很大的作用。

EPEA\* 改进的也是分支因子指数增长的缺陷。EPEA\* 和 A\* 的不同之处在于, 在扩展一个状态节点时, EPEA\* 生成的后继状态节点不完整, 也就是只将一部分的后继状态节点加入 open-list 中。文献<sup>[10]</sup>证明了 OD 是 EPEA\* 的特例, EPEA\* 的实现细节可以参考文献<sup>[11]</sup>。

ID 改进了 A\* 搜索状态空间的指数增长缺陷, 其本质上是一种智能体之间的独立性检测。ID 尝试将包含  $k$  个智能体的多智能体路径规划问题分解成多个包含更少智能体的子问题。在开始阶段, 每个智能体不考虑其他智能体, 各自计算自己的最优路径, 然后对计算出的  $k$  条最优路径进行独立性检测, 如果两组路径之间发生了冲突, 就将对应的智能体合并为一组, 不断地进行合并分组, 直到剩下的智能体分组对应的路径组之间不存在冲突, 检测终止。使用 ID 的 A\* 在最坏情况下等于朴素的 A\*, 当不是最坏情况时, ID 将原来包含  $k$  个智能体的多智能体路径规划问题成功分解, 大幅提高了 A\* 算法的执行效率。ID 在多智能体路径规划中是一种常用的方法, 可以提高多数多智能体路径规划算法的效率。

M\* 同时改进了状态空间的指数增长和分支因子的指数增长缺陷, 其核心思想和 ID 类似。M\* 大幅提高了 A\* 算法的性能上限, 可以较快地求解规模为 20 个左右智能体的路径规划问题。M\* 中存在一个重要的数据结构——冲突集合, 冲突集合初始为空, 当开始扩展状态节点时, M\* 中所有智能体依照不考虑其他智能体的各自的最优路径行动, 这意味着搜索状态空间的维度是 1, 分支因子的维度也是 1。这样的扩展会使  $q \geq 2$  个智能体在节点  $v$  发生冲突时中止。此时, 发生冲突的这  $q$  个智能体会被加入到冲突集合中, 然后重启整个搜索过程。在搜

索重启时,冲突集中的 $q$ 个智能体会执行朴素的多智能体A\*搜索,其余的 $k-q$ 个智能体依旧按照各自的最优路径参与全局状态扩展,这时状态空间的维度是 $q$ ,分支因子为 $b^q$ 。M\*在所有智能体到达目标位置时终止,M\*的最坏情况仍然是回归到朴素的A\*。M\*的改进型rM\*[12]将冲突的智能体分割为没有冲突的分组,递归地解决生成的子问题,从而加速了算法执行。另一种改进型ODrM\*[13]在搜索层面上采用ODA\*取代朴素的A\*,将A\*类最优算法求解问题的规模扩大到30个智能体。

上述A\*的改进方法之间并非互斥,所有方法都可以同时使用。这些方法仍然是在k-agent状态空间中求解,这也是这类方法的一个显著特征。

## 2.2 代价增长树搜索

与基于A\*的算法不同,代价增长树搜索[14]没有直接在k-agent状态空间中进行搜索。代价增长树搜索分为高层次搜索和低层次搜索2层。高层次搜索的目的是找到每个智能体 $a_i$ 在问题的最优解中所花费的代价 $c_i$ ,即 $(c_1, c_2, \dots, c_k)$ 。低层次搜索的目的是验证对于给定的 $(c_1, c_2, \dots, c_k)$ ,是否存在一组最优解 $(\pi_1, \pi_2, \dots, \pi_k)$ ,使得 $\forall i \in \{1, 2, \dots, k\}, \pi_i = c_i$ 。

### 2.2.1 高层次搜索

高层次搜索是在代价增长树上进行搜索。代价增长树的状态节点是一组 $k$ 维向量 $(c_1, c_2, \dots, c_k)$ ,代表每个智能体 $a_i$ 从起始位置 $s_i$ 到目标位置 $g_i$ 花费代价为 $c_i$ 的所有可能的路径集合。代价增长树的根节点是 $(o_1, o_2, \dots, o_k)$ , $o_i$ 是智能体 $a_i$ 不考虑其他智能体的最优路径的代价。 $(o_1, o_2, \dots, o_k)$ 是全局最优解的下限。代价增长树通过选取当前状态节点中的一个智能体 $a_i$ ,令 $c_i + 1$ ,生成后继的状态节点。对于代价搜索树的一个状态节点 $(c_1, c_2, \dots, c_k)$ ,如果通过低层次搜索的验证,那么 $(c_1, c_2, \dots, c_k)$ 就是一个可行解。因为代价增长树中同一层次的状态节点的代价总和相等,所以在高层次搜索中可以采用宽度优先搜索来找到全局最优解。因此,当找到第一个通过低层次搜索验证的状态节点 $(c_1, c_2, \dots, c_k)$ 时, $(c_1, c_2, \dots, c_k)$ 就是全局最优解,算法终止。

### 2.2.2 低层次搜索

低层次搜索是对高层次搜索的状态节点 $(c_1, c_2, \dots, c_k)$ 的验证。低层次搜索首先计算每个智能体 $a_i$ 从起始位置 $s_i$ 到目标位置 $g_i$ 花费代价为 $c_i$ 且不考虑其他智能体的所有路径集合,分别使用MDD[15]存储。所有MDD的交叉乘积结果就是所有满足 $(c_1, c_2, \dots, c_k)$ 的全局路径集合,这其实是k-agent状态空间中的一个子空间。低层次搜索对MDD叉乘的结果进行搜索,寻找可行解。低层次搜索是验证的过程而非最优化的过程,因此,可以采用有界深度优先搜索来实现。

### 2.2.3 剪枝加速

代价增长树搜索存在的一个问题是不能很快地验证给定状态节点是否存在解。这个问题可以通过

在低层次搜索中验证其子问题是否存在可行解来改进。常用的一种策略是选取所有的智能体对 $(a_i, a_j)$ , $i, j \in \{1, 2, \dots, k\}, i \neq j$ ,当某对智能体 $(a_i, a_j)$ 之间不存在无冲突的路径组合时,低层次搜索就可以返回 $(c_1, c_2, \dots, c_k)$ 无解。

## 2.3 基于冲突的搜索

基于冲突的搜索(CBS)[16]和代价增长树搜索有些相似,同样分为高层次搜索和低层次搜索,不同之处在于,CBS求解的是一系列的单智能体路径规划问题。

### 2.3.1 约束与一致路径

在基于冲突的搜索中,智能体会受到一些约束。对智能体 $a_i$ 的一组约束可以表示为 $\langle a_i, v, t \rangle$ ,表示智能体 $a_i$ 在 $t$ 时刻不能处于位置 $v$ 。智能体 $a_i$ 的一致路径指的是满足关于 $a_i$ 的所有约束的路径。全局可行解中每个智能体的路径必须是一致路径,反之则不成立,原因是无法保证智能体的一致路径之间不存在冲突。

### 2.3.2 高层次搜索

基于冲突的搜索的高层次搜索是在约束树上进行的。约束树是一棵二叉树,树上的每个状态节点包含一个约束集合、一组智能体的一致路径(每个智能体对应一条一致路径)和目前为止的全局代价。约束树根节点中的约束集合是空集。CBS中当前状态节点的后继状态节点会继承当前状态节点的约束集合并加入对某个智能体的一组新约束。一组智能体的一致路径只有在任意一组路径之间都不存在冲突时才是全局可行解。

对于约束树中的某个状态节点,低层次搜索为其寻找一组智能体的一致路径,然后按照时间步模拟每个智能体的行动。如果整个模拟过程中没有发生冲突,那么当前状态节点就是目标状态节点;如果发生了冲突,就需要进行分割操作。分割操作指当约束树中的某个状态节点中的一致路径包含一个冲突 $\langle a_i, a_j, v, t \rangle$ 时,需要生成2个约束 $\langle a_i, v, t \rangle$ 和 $\langle a_j, v, t \rangle$ 。对于这2个约束,当前状态节点会分别生成2个后继状态节点,它们的约束集合相比于当前状态节点的约束集合分别增加了 $\langle a_i, v, t \rangle$ 和 $\langle a_j, v, t \rangle$ 。2个后继状态节点将继承当前状态节点中新加入约束中的智能体之外的其他智能体的一致路径,拥有新约束的智能体将会通过低层次搜索生成新的一致路径。

CBS的高层次搜索一般采用最优优先搜索,优先扩展全局代价最小的状态节点。

### 2.3.3 低层次搜索

CBS的低层次搜索根据约束树节点中的约束集合,寻找满足每个智能体相关约束的一组一致路径。但是低层次搜索无法保证一致路径之间不存在冲突。在低层次搜索中,所有的单智能体路径规划算法都适用。

### 2.3.4 改进的基于冲突搜索

朴素的CBS存在节点分割低效和重复计算较多等问题。为了解决这些问题,研究人员对CBS进行了很多的改进和扩展。Meta-agent CBS[16]加入了合并

操作, 合并操作将冲突中的 2 个智能体合并为一个新的智能体 meta-agent, 而非通过分割操作为两者添加新的约束。在搜索中设置智能体之间的关联阈值, 对超过阈值的冲突采用合并操作, 低于阈值的冲突采用分割操作。ICBS<sup>[17]</sup> 在存在多个冲突时更智能地选取某个冲突进行合并或者分割, 以减小约束树的大小。除此以外, ICBS<sup>[17]</sup> 还提出在执行合并操作之后可以重启整个搜索, 按照新的智能体组成来搜索, 以减少约束树中状态节点的重复计算。文献[18]提出当发生某些冲突时, 可以先尝试寻找冲突中的智能体是否存在绕过当前冲突位置并且和当前一致路径的代价相等的其他一致路径, 如果存在, 只需更新约束树中当前状态节点的一致路径, 以此减小约束树的规模。HCBS<sup>[19]</sup> 为高层次搜索加入了新的启发式函数, 能够更好地对约束树进行剪枝。文献[20]提出了新的约束, 正约束  $\langle a_i, v, t \rangle$  表示智能体  $a_i$  在  $t$  时刻必须处于位置节点  $v$ , 负约束  $\langle a_i, v, t \rangle$  表示智能体  $a_i$  在  $t$  时刻不能占据位置节点  $v$ 。在此基础上, 文献[20]修改了分割操作的分割方式, 不再对称地分割为 2 个负约束  $\langle a_i, v, t \rangle$  和  $\langle a_j, v, t \rangle$ , 而是将冲突分割为一个正约束  $\langle a_i, v, t \rangle$  和一个负约束  $\langle a_j, v, t \rangle$ 。

在加入了上述这些对 CBS 的优化之后, CBS 能够求解 30 个 ~ 60 个智能体的路径规划问题。基于冲突的搜索是多智能体路径规划领域中比较新颖的方法, 也是目前研究的热点。

#### 2.4 基于规约的方法

上述 3 类最优多智能体路径规划算法本质上都是基于搜索的算法, 而基于规约的方法与它们不同。因为多智能体路径规划问题是 NP-hard 的问题, 可以将多智能体路径规划问题规约为其他的标准问题, 如 SAT、CSP 和 ASP 等。完成规约的正确性证明后, 便可以利用这些问题已有的高质量求解器来求解多智能体路径规划问题。

文献[21]将多智能体路径规划问题规约成了 CSP 问题, 但是该过程对问题中的地图有所要求, 只有地图可以被抽象为已知的子图, 如环和团等形状时, 规约才是正确的。文献[22]将多智能体路径规划问题规约为 SAT 问题, 将问题中地图的结构、智能体的位置和约束编码为布尔变量, 然后用这些布尔变量生成标准的 SAT 问题来验证是否存在全局

代价为  $C$  的可行解, 遍历所有可能的全局代价  $C$  就能够生成全局最优解。文献[23]通过改进问题的编码方式, 成为较早能够解决求和代价函数的多智能体路径规划问题的 SAT 求解器。文献[24]将 ID 用于基于 SAT 的方法中来求解问题。文献[25]提出新的代价估计方法并减少 SAT 模型中变量的数量, 从而加速了算法执行。文献[26]通过将多智能体路径规划问题中 2 个智能体之间的约束转化为 ASP 中的程序  $P$  来求解问题,  $P$  的答案集就是问题的解集。文献[27-28]将多智能体路径规划问题规约为一个网络流问题, 规约后网络流中流的深度和多智能体路径规划中的时间步相关, 然后将多智能体路径规划问题的形式转变为一组等式和一个目标函数, 利用整数线性规划来求解最优解。文献[29]将多智能体路径规划问题规约为 TSP 问题, 改进了 IPL 模型, 提出了有效的模型绩效评价指标, 利用已有的 ILP 求解器求解变形之后的问题。

基于规约的方法的难点在于规约的证明, 一般需要极强的数理功底。在一般情况下, 这类方法的求解速度高于上述基于搜索的最优算法, 但是它们无法保证完成规约证明之后使用相应求解器时的高效性。

### 3 近似的多智能体路径规划算法

因为多智能体路径规划问题是 NP-hard 问题, 求解最优解的算法需要很长的执行时间, 为了加速问题的求解, 往往需要牺牲一些结果的最优性。近似的多智能体路径规划算法可以分为无边界次优的和有边界次优的算法。多数近似算法都是无边界次优的, 而有边界的次优算法一般都是最优算法的衍生算法。一般来说, 有边界的次优算法的速度会快于最优算法, 但是慢于无边界次优的算法; 在结果的最优性上, 有边界的次优算法会略逊于最优算法, 但是优于无边界的次优算法。

#### 3.1 无边界次优的多智能体路径规划算法

无边界次优的多智能体路径规划算法能够快速计算出可行解, 但是无法保证结果的质量。早期研究的局限性都较大, 比如文献[30]能够保证算法具备完整性, 但是结果的质量可能远低于最优结果, 并且在每个时间步中只允许一个智能体采取行动。无边界次优算法分类与对比如表 3 所示。

表 3 各类无边界次优的多智能体路径规划算法对比

Table 3 Comparison of each kind of unbounded suboptimal MAPP algorithms

算法分类	优点	缺点	适用问题的特点
基于搜索的算法	易于实现, 结果质量一般较好	求解速度较慢, 一般无法保证完整性	适用于智能体数量较少的问题, 在智能体密度较低时效果较好
基于规则的算法	求解速度很快	求解的结果可能比最优解差很多, 只有在特定的地图上才有完整性	能够求解大规模智能体问题, 但是在对结果质量要求较高时不适用
基于规约的算法	可以利用已有的一些求解算法	规约的证明比较困难	规约的模型决定算法适合求解问题的类型
其他算法	求解结果的质量较好	无法保证完整性	能够求解大规模智能体问题



### 3.1.1 基于搜索的无边界次优算法

文献[31]提出了 CA\* 及其改进型 HCA\* 和 WHCA\*。这类算法顺序地为每个智能体单独规划路径,已经规划的路径会被记录在留存表中,后续在智能体规划路径时不能和留存表中的路径发生冲突,但上述算法并不具备完整性。文献[32]提出了一种具备完整性的无边界次优算法,其本质上是对 A\* 算法的约束放松。文献[33]将状态空间进行抽象化来减少启发式函数的计算时间。文献[34]提出了一种动态修改路径的 2 段协调方法,在第 1 阶段使用 A\* 搜索每个智能体的最优路径,在第 2 阶段采用增量式 A\* 算法解决路径之间的冲突问题。文献[35]在文献[31]的基础上,将窗口放置于已知的冲突位置附近,并将所有智能体根据参与到冲突中的可能性大小进行优先级排序,按照优先级来为智能体规划路径。文献[36]提出了一种基于保留区域的分布式多智能体路径规划算法,解决了多智能体路径之间高度耦合的问题。一般而言,与基于规则的算法相比,基于搜索的无边界次优算法的速度没有特别大的优势,但是结果的质量会更好。

### 3.1.2 基于规则的无边界次优算法

在基于规则的无边界次优算法中,智能体在不同的场景下有不同的行动规则,并且不需要进行大规模的搜索。基于规则的无边界次优算法能够很快地求解问题,但是其结果质量可能远逊于最优解。

基于规则的无边界次优算法的一个重要性质就是完整性,但是这类算法的完整性往往对问题中地图的性质有特殊的要求。文献[37]提出一种基于规则且具备完整性的无边界次优算法,但是其实际实现时难度极大。文献[38]算法只有在树形图上才具备完整性。文献[39]算法只有在双连通图上才具备完整性。文献[40]算法只有在强双连通图上才具备完整性。

文献[41-43]只有在地图中至少存在 2 个未被智能体占据的空位置节点时才具备完整性。文献[41]引入 2 个 macro 操作符,macro 可以将智能体推到空位置节点,还可以交换 2 个智能体的位置。文献[42]提出文献[41]的改进算法,不同于文献[41]中每个时间步内只允许一个智能体行动,文献[42]可以并行地为所有智能体规划路径。文献[43]通过引入新的旋转操作符来改进文献[41]算法。

此外,还存在一些混合类型的算法,即算法中既存在智能体行动的规则也存在大规模的搜索,比如文献[44]通过引入交通规则来减少搜索。目前,对混合方法的研究较少,但是这类混合方法给多智能体路径规划的研究带来了很大的启发。

上述基于规则的无边界次优算法的求解速度一般较快,但是无法保证其结果的质量。

### 3.1.3 基于规约的无边界次优算法

文献[45]通过将问题规约为 CSP 问题,即令每个智能体对应一个变量,变量的值域就是智能体从

起始位置到目标位置的所有可能路径的集合,则问题的约束就是智能体的路径之间不能存在冲突。文献[46]通过使用矩阵计算加速变量值域的缩减,引入动态变量排序和快速随机重启等技巧,取得了较好的效果。文献[47]提出了一种基于 ASP 的无边界近似分布式多智能体路径规划算法,其能够处理很大规模的多智能体路径规划问题。文献[48]提出基于 SAT 的无边界次优算法,其通过移除文献[23]中的数量约束,从而将文献[23]算法从最优算法转变为无边界次优的算法。

### 3.1.4 其他无边界次优算法

其他无边界次优算法主要包括基于强化学习、基于遗传算法、基于协同进化和基于蚁群算法的算法。

国内近期提出了一类基于强化学习的多智能体路径规划算法。文献[49]通过采用无模型的在线 Q 学习方法,使得多个智能体重复“探索-学习-利用”过程,积累历史经验评估动作策略并优化决策。文献[50]提出了一种基于分层强化学习和人工势场的多智能体路径规划算法,其将多智能体的所在环境虚拟为一个人工势能场,根据先验知识确定每个节点的势能值,代表最优策略可获得的最大回报,利用分层强化学习方法中的无环境模型学习和局部更新能力,将策略更新过程限制在规模较小的局部空间或维度较低的高层空间中。

在早期的基于遗传算法中,文献[51]采用链接图法建立了智能体工作空间模型,采用遗传算法规划多智能体运动路径,引入适应值调整矩阵的新概念,实现了对多智能体路径的全局优化。近期,文献[52]以多智能体执行任务的自身代价与任务间的关联代价为优化目标,使用遗传算法求解问题。

在基于协同进化的算法中,文献[53]通过采用协同进化算法、设计适应度评价函数和引入一系列新的变异操作算子,以求解问题。

在基于蚁群的算法中,文献[54]通过建立有权图模型,采用蚁群算法求解多智能体路径规划问题。文献[55]提出了一种多步长的改进蚁群算法,以改进文献[54]算法收敛速度慢、容易陷入局部最优的缺陷。

上述算法的主要问题是容易陷入局部最优,并且无法保证结果的质量与算法的完整性。

## 3.2 有边界次优的多智能体路径规划算法

有边界次优的多智能体路径规划算法指对于常量  $\varepsilon > 0$ , 算法给出结果的代价最多是最优结果代价的  $1 + \varepsilon$  倍,一般称这类算法为  $\varepsilon$  次优的算法。理论上而言,当  $\varepsilon$  变大,算法的速度也会变快。因此,在算法的速度和结果质量上存在均衡。现有有边界次优的多智能体路径规划算法都是在前述最优算法的基础上牺牲最优性得来的次优算法。4 种主要的有边界次优算法的对比结果如表 4 所示。

表 4 有边界次优的多智能体路径规划算法对比  
Table 4 Comparison of bounded suboptimal MAPP algorithms

算法分类	引入次优性的方式	优缺点	求解问题规模
A * 类算法	修改启发式函数	求解速度最慢, 但是易于实现	针对 60 个 ~ 120 个智能体的问题时效果较好
基于代价增长树搜索的算法	不存在经典问题下的有边界次优算法	—	—
基于冲突的搜索类算法	在高层次搜索和低层次搜索中都可以修改启发式函数	需要根据经验来确定合并的阈值	目前在 120 个 ~ 200 个智能体的问题中比较适用
基于规约的算法	由规约后问题的有边界次优算法引入次优性	规约证明困难, 优缺点由规约后采用的算法决定	由规约后采用的算法决定

### 3.2.1 基于 A \* 的有边界次优算法

多数基于 A \* 的有边界次优算法都是通过修改 A \* 搜索的启发式函数来实现的。文献[56]通过使用  $g + (1 + \varepsilon)h$  的启发式函数来选择待扩展的状态节点。前述所有基于 A \* 的最优算法都可以使用这种估计函数转变为有边界次优算法, 比如文献[12]提出的 inflated M \*。

文献[57]使用  $(C - g)/h$  的启发式函数, 并且将算法次优的边界改进为问题要求的次优边界与 open-list 中最小的启发式函数值的积。

### 3.2.2 基于代价增长树搜索的有边界次优算法

因为代价增长树搜索的高层次搜索是宽度优先搜索, 低层次搜索只是高层次搜索的验证, 不存在启发式函数, 所以对于基本的多智能体路径规划问题不存在有边界次优的算法。但是如果在多智能体路径规划问题的定义中加入新的约束, 比如保证问题中地图的所有边权值不同, 则文献[58]算法可以看作是代价增长树搜索的一个有边界次优衍生算法。

### 3.2.3 基于冲突搜索的有边界次优算法

CBS 的低层次搜索通常是最优的单智能体最短路径算法, 比如 A \*。因此, 所有基于 A \* 的有边界次优算法都可以在 CBS 的低层次搜索中实现, 这是实现基于 CBS 的有边界次优算法的一种途径。

如果要在高层次搜索中引入次优性, 比较常用的方法是引入 FS (Focal Search)<sup>[59]</sup>。FS 在每次扩展时从 focal-list 中选择状态节点, 而不是从 open-list 中选择。对于常数  $\omega$ , focal-list 中存储的是 open-list 中代价小于等于  $\omega$  倍 open-list 中最优状态节点代价的状态节点。因此, 在引入 FS 之后, 高层次搜索就有 2 个启发式函数可以引入次优性, 即 open-list 的启发式函数和 focal-list 的启发式函数。

文献[60]在 CBS 的高层次搜索和低层次搜索中引入了次优性, 为  $\omega_1$  次优。文献[61]算法是文献[60]算法的改进, 其加入了直连边, 使得算法是  $\omega_1\omega_2$  次优。文献[62]算法在文献[61]的基础上进行优化, 使得算法仍然是  $\omega_1$  次优。

### 3.2.4 基于规约的有边界次优算法

在多智能体路径规划算法中, 单独研究基于规约的有边界次优算法的工作较少, 因为只要证明了

规约的正确性, 那么相关规约的有边界次优算法都可以应用于多智能体路径规划问题的求解中。比如, 文献[48]通过放松文献[23]中的数量约束, 赋予其更松弛的代价边界, 将最优算法转变为有边界次优的算法。

## 4 研究展望

近年来, 传统的基于搜索的算法虽然仍然是研究热点, 但是基于规约的方法以及其他方法也引起了学者们的关注, 比如文献[63]使用深度强化学习和卷积神经网络来求解多智能体路径规划问题, 文献[64]将基于搜索的方法和基于 SAT 的方法相结合, 以求解多智能体路径规划问题。值得注意的是, 现有所有多智能体路径规划算法都不能在所有类型的问题中取得超过其他所有算法的性能, 即不同算法适用的问题类型不同。本文将未来的研究方向归纳为以下 5 个方面:

1) 多智能体路径规划理论需要系统地对问题中相应参数(比如智能体数量、智能体密度和地图尺寸等)对问题求解难度的影响进行量化分析, 上述研究可以采用控制变量法来实现。在相关参数中, 本文认为智能体密度和地图中的障碍物密度可能是影响问题求解的主要因素。

2) 新算法逐渐涌现, 但是对于目标函数之间的差异, 目前研究比较少, 多数研究都只针对其中一种来验证算法的性能。在前述的 4 种目标函数中, 求和形式的目标函数在研究中被广泛应用, 但是求最大值形式的目标函数才是最具有实际应用价值的, 这就使得在理论研究和实际应用中存在鸿沟。求和形式的目标函数在研究中最受欢迎, 是因为该类型的目标函数容易求导并证明算法的最优性, 这也意味着部分最优的算法可能无法适用最值型的目标函数。因此, 需要对不同目标函数在不同算法中的兼容性进行研究。

3) 不同算法擅长求解的问题一般不同, 需要对已有算法之间的关联性和性能差异做出量化分析, 目前也缺少被广泛接受的标准测试集。由于多智能体路径规划算法的开源较少, 且论文一般不会提供相关代码, 在算法之间进行比较的工作量会非常大。

4)对于经典多智能体路径规划问题的变形进行研究正在逐渐受到重视。经典模型对现实生活中问题的建模能力比较弱,在实际生活中,路径规划问题可能会包含信号灯、优先级、概率等因素,使得经典算法不再适用。因此,为了更好地在实际问题中应用多智能体路径规划算法,主流的研究不能只局限于经典模型。

5)类似于机器学习中的模型融合,考虑到各类多智能体路径规划算法对不同特点问题的求解性能差异,可以对各类算法的融合进行研究。本文认为,多智能体路径规划算法之间的融合主要有2种方法:第1种方法是对于给定的问题,可以利用多线程使用各种算法来求解,当某种算法求解成功之后终止所有算法,但是,考虑到这些算法对计算资源的需求较大,融合算法的计算代价可能极高;第2种方法是在求解问题之前,先分析问题以获得问题参数,然后根据参数来选取求解算法,问题中的部分参数如智能体数量、地图尺寸等一般已知,但是在问题中还会存在一些未知的参数,如智能体密度、障碍物密度等,对于这类参数可以通过采样来估计。在获得上述参数之后,通过建立决策树模型来决定采取何种算法求解问题。

## 5 结束语

本文介绍多智能体路径规划问题以及最优算法、无边界的次优算法和有边界的次优算法3类求解算法。最优算法能够保证结果的最优性,但是速度最慢,无边界的次优算法速度最快,但是结果质量无法保证,有边界的次优算法能够保证结果的质量,速度处于最优算法与无边界的次优算法之间。在实际应用中,只有当问题规模较小时,才适合使用最优算法,而当问题规模较大时,可以根据对结果质量的需求使用有边界或无边界的次优算法。鉴于不同算法适用问题的类型不同,下一步除了探究性能更好的算法之外,还将深入分析已有算法适用问题的差异性,量化问题参数对规划结果影响的大小,推动融合算法的进一步发展。

### 参考文献

- [1] TANG Yong, HE Donglin, ZHU Xiping. Aircraft taxiing path planning based on multi-agent system[J]. Journal of Jiangsu University(Natural Science Edition), 2019(5): 559-565. (in Chinese)  
唐勇,何东林,朱新平.基于多智能体系统的飞机滑行路径规划[J].江苏大学学报(自然科学版),2019(5): 559-565.
- [2] WURMAN P R, D' ANDREA R, MOUNTZ M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses [C]//Proceedings of the 22nd AAAI Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2007: 56-78.
- [3] DRESNER K M, STONE P. A multiagent approach to autonomous intersection management [J]. Journal of Artificial Intelligence Research, 2008, 31(3): 591-656.
- [4] VELOSO M, BISWAS J, COLTIN B, et al. CoBots: robot symbiotic autonomous mobile service robots [C]//Proceedings of International Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2015: 566-574.
- [5] HUANG Jin, HUANG Zongwen, LING Ziyang. Application of multi-agent road finding system in computer games[J]. Computer Knowledge and Technology, 2012, 8(13): 3159-3164. (in Chinese)  
黄进,黄宗文,凌子燕.多智能体寻路系统在计算机游戏上的应用[J].电脑知识与技术,2012,8(13): 3159-3164.
- [6] FU Mengjia, YOU Xiaoming. A survey of multi robot system and its path planning methods [J]. Software Guide, 2017, 16(1): 177-179. (in Chinese)  
付梦家,游晓明.多机器人系统及其路径规划方法综述[J].软件导刊,2017,16(1): 177-179.
- [7] SURYNEK P. An optimization variant of multi-robot path planning is intractable [C]//Proceedings of the 24th AAAI Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2010: 67-89.
- [8] RYAN M R K. Exploiting subgraph structure in multi-robot path planning [J]. Journal of Artificial Intelligence Research, 2008, 31: 497-542.
- [9] STANDLEY T. Finding optimal solutions to cooperative pathfinding problems [C]//Proceedings of the 24th AAAI Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2010: 173-178.
- [10] GOLDENBERG M, FELNER A, STURTEVANT N R, et al. Optimal-generation variants of EPEA [EB/OL]. [2019-11-02]. <https://webdocs.cs.ualberta.ca/~holte/Publications/SoCS2013-EPEA.pdf>.
- [11] GOLDENBERG M, FELNER A, STERN R, et al. Enhanced partial expansion A\* [J]. Journal of Artificial Intelligence Research, 2014, 50(1): 141-187.
- [12] WAGNER G, CHOSET H. Subdimensional expansion for multirobot path planning [EB/OL]. [2019-11-02]. [https://www.ri.cmu.edu/pub\\_files/2015/2/subdim\\_journal.pdf](https://www.ri.cmu.edu/pub_files/2015/2/subdim_journal.pdf).
- [13] FELNER C, WAGNER G, CHOSET H. ODrM\* optimal multirobot path planning in low dimensional search spaces [C]//Proceedings of 2013 IEEE International Conference on Robotics and Automation. Washington D. C., USA: IEEE Press, 2013: 56-89.
- [14] SHARON G, STERN R, GOLDENBERG M, et al. The increasing cost tree search for optimal multi-agent pathfinding [J]. Artificial Intelligence, 2013, 195: 470-495.
- [15] SRINIVASAN A, HAM T, MALIK S, et al. Algorithms for discrete function manipulation [C]//Proceedings of IEEE International Conference on Computer-aided Design. Washington D. C., USA: IEEE Press, 1990: 78-90.
- [16] SHARON G, STERN R, FELNER A, et al. Conflict-based search for optimal multi-agent pathfinding [J]. Artificial Intelligence, 2015, 219: 40-66.
- [17] BOYARSKI E, FELNER A, STERN R, et al. ICBS: improved conflict-based search algorithm for multi-agent pathfinding [C]//Proceedings of International Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2015: 67-90.



- [18] BOYARSKI E, FELNER A, SHARON G, et al. Don't split, try to work it out: bypassing conflicts in multi-agent pathfinding [C]//Proceedings of International Conference on Automated Planning and Scheduling. Washington D. C. , USA; IEEE Press, 2010:47-51.
- [19] FELNER A. Adding heuristics to conflict-based search for multi-agent path finding [EB/OL]. [2019-11-02]. <http://idm-lab.org/bib/abstracts/papers/icaps18a.pdf>.
- [20] LI J, HARABOR D, STUCKEY P, et al. Disjoint splitting for multi-agent path finding with conflict-based search [C]//Proceedings of International Conference on Automated Planning and Scheduling. Washington D. C. , USA; IEEE Press, 2019:279-283.
- [21] RYAN M. Constraint-based multi-robot path planning [C]//Proceedings of International Conference on Robotics and Automation. Washington D. C. , USA; IEEE Press, 2010:38-54.
- [22] SURYNEK P. Towards optimal cooperative path planning in hard setups through satisfiability solving [C]//Proceedings of the Pacific Rim International Conferences on Artificial Intelligence. Washington D. C. , USA; IEEE Press, 2012:564-576.
- [23] SURYNEK P, FELNER A, STERN R, et al. Efficient SAT approach to multi-agent path finding under the sum of costs objective [C]//Proceedings of ECAI-European Conference on Artificial Intelligence. Berlin, Germany: Springer, 2016:810-818.
- [24] SURYNEK P, JIRI S, FELNER A, et al. Integration of independence detection into SAT-based optimal multi-agent path finding—a novel sat-based optimal MAPF solver [C]//Proceedings of the 9th International Conference on Agents and Artificial Intelligence. Berlin, Germany: Springer, 2017:116-136.
- [25] BARTÁK R, ŠVANCARA J. On SAT-based approaches for multi-agent path finding with the sum-of-costs objective [EB/OL]. [2019-11-02]. [http://svancara.net/files/socs\\_soc.pdf](http://svancara.net/files/socs_soc.pdf).
- [26] ERDEM E, KISA D G, OZTOK U, et al. A general formal framework for pathfinding problems with multiple agents [C]//Proceedings of AAAI Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2013:290-296.
- [27] YU J, LAVALLE S M. Planning optimal paths for multiple robots on graphs [EB/OL]. [2019-11-02]. <https://arc.cs.rutgers.edu/files/YuLav13ICRA.pdf>.
- [28] YU J, LAVALLE S M. Optimal multirobot path planning on graphs: complete algorithms and effective heuristics [J]. IEEE Transactions on Robotics, 2016, 32(5):1-15.
- [29] SHU Xiangxiang. Research on optimal path planning of multi robots [J]. China Computer and Communication, 2017(15):62-64. (in Chinese)  
舒翔翔. 多机器人最优路径规划研究 [J]. 信息与电脑, 2017(15):62-64.
- [30] SPIRALIS D K G M P. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications [EB/OL]. [2019-11-02]. <https://www.computer.org/csdl/proceedings-article/focs/1984/0715921/12OmNC3FGi5>.
- [31] SILVER D. Cooperative pathfinding [C]//Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference. [S. l.]: AAAI Press, 2005:78-90.
- [32] STANDLEY T S, KORF R E. Complete algorithms for cooperative pathfinding problems [C]//Proceedings of the 22nd International Joint Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2011:668-673.
- [33] STURTEVANT N R, BURO M. Improving collaborative pathfinding using map abstraction [EB/OL]. [2019-11-02]. <https://www.aaai.org/Papers/AIIDE/2006/AIIDE06-017.pdf>.
- [34] ZHAO Yongsheng, SUN Wenlei. Multi robot path planning with dynamic path modification [J]. Mechanical Science and Technology for Aerospace, 2018, 37(10):13-18. (in Chinese)  
晁永生, 孙文磊. 动态修改路径的多机器人路径规划 [J]. 机械科学与技术, 2018, 37(10):13-18.
- [35] BNAYA Z, FELNER A. Conflict-oriented windowed hierarchical cooperative A\* [C]//Proceedings of IEEE International Conference on Robotics and Automation. Washington D. C. , USA; IEEE Press, 2014:56-78.
- [36] CAO Qixin, HUANG Xianqun, ZHU Xiaoxiao, et al. Path planning of distributed multi robot based on reserved area [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2018, 46(12):76-81. (in Chinese)  
曹其新, 黄先群, 朱笑笑, 等. 基于保留区域的分布式多机器人路径规划 [J]. 华中科技大学学报(自然科学版), 2018, 46(12):76-81.
- [37] KORNHAUSER D, MILLER G, SPIRAKIS P. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications [C]//Proceedings of the 25th Annual Symposium on Foundations of Computer Science. Washington D. C. , USA; IEEE Press, 2002:998-1024.
- [38] KHORSHID M M, HOLTE R C, STURTEVANT N. A polynomial-time algorithm for non-optimal multi-agent pathfinding [EB/OL]. [2019-11-02]. <http://webdocs.cs.ualberta.ca/~holte/Publications/SoCS11Multiagent.pdf>.
- [39] SURYNEK P. A novel approach to path planning for multiple robots in bi-connected graphs [C]//Proceedings of IEEE International Conference on Robotics and Automation. Washington D. C. , USA; IEEE Press, 2009:678-690.
- [40] BOTE A, SURYNEK P. Multi-agent path finding on strongly biconnected digraphs [C]//Proceedings of the 29th AAAI Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2015:56-80.
- [41] LUNA R, BEKRIS K E. Efficient and complete centralized multi-robot path planning [C]//Proceedings of the 14th Annual Symposium on Combinatorial Search. Washington D. C. , USA; IEEE Press, 2011:899-920.
- [42] SAJID Q, LUNA R, BEKRIS K E. Multi-agent path finding with simultaneous execution of single-agent primitives [EB/OL]. [2019-11-02]. [http://www.ryanluna.com/papers/Sajid\\_SoCS\\_12.pdf](http://www.ryanluna.com/papers/Sajid_SoCS_12.pdf).
- [43] DEWILDE B, MORS A W T, WITTEVEEN C. Push and rotate: a complete multi-agent pathfinding algorithm [J]. Journal of Artificial Intelligence Research, 2014, 51:443-492.
- [44] WANG K H C, BOTE A. Fast and memory-efficient multi-agent pathfinding [C]//Proceedings of International Conference on Automated Planning and Scheduling. Washington D. C. , USA; IEEE Press, 2008:380-387.

- [45] JANSEN M R, STURTEVANT N R. A new approach to cooperative pathfinding [C]//Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. Washington D. C., USA: IEEE Press, 2008:12-16.
- [46] WANG J, LI J, MA H, et al. A new constraint satisfaction perspective on multi-agent path finding: preliminary results [C]//Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems. Washington D. C., USA: IEEE Press, 2019:2253-2255.
- [47] PIANPAK P, SON T C, TOUPS Z O, et al. A distributed solver for multi-agent path finding problems [C]//Proceedings of the 1st International Conference on Distributed Artificial Intelligence. New York, USA: ACM Press, 2019:2-10.
- [48] SURYNEK P, FELNER A, STERN R, et al. Modifying optimal SAT-based approach to multi-agent path-finding problem to suboptimal variants [C]//Proceedings of the 20th Annual Symposium on Combinatorial Search. Washington D. C., USA: IEEE Press, 2017:25-43.
- [49] WANG Yiran, JING Xiaochuan, TIAN Tao, et al. Multi-Agent path planning based on reinforcement learning [J]. Computer Application and Software, 2019, 36(8):165-171. (in Chinese)  
王毅然, 经小川, 田涛, 等. 基于强化学习的多 Agent 路径规划方法研究 [J]. 计算机应用与软件, 2019, 36(8):165-171.
- [50] ZHENG Yanbin, LI Bo, AN Deyu, et al. Multi-Agent path planning method based on hierarchical reinforcement learning and artificial potential field [J]. Computer Application, 2015, 35(12):3491-3496. (in Chinese)  
郑延斌, 李波, 安德宇, 等. 基于分层强化学习及人工势场的多 Agent 路径规划方法 [J]. 计算机应用, 2015, 35(12):3491-3496.
- [51] SUN Shudong, LIN Mao. Coordinated path planning of multiple mobile robots based on genetic algorithm [J]. Acta Automatica Sinica, 2000, 26(5):672-676. (in Chinese)  
孙树栋, 林茂. 基于遗传算法的多移动机器人协调路径规划 [J]. 自动化学报, 2000, 26(5):672-676.
- [52] FAN Yuan, LI Wenfeng, HE Lijun. Cooperative scheduling of intelligent storage multi mobile robot based on improved genetic algorithm [J]. Journal of Wuhan University of Technology (Information and Management Engineering), 2019, 41(3):293-298, 311. (in Chinese)  
范媛, 李文锋, 贺利军. 基于改进遗传算法的智能仓储多移动机器人协同调度 [J]. 武汉理工大学学报(信息与管理工程版), 2019, 41(3):293-298, 311.
- [53] LEI Xiaoyu, YANG Shengyue, ZHANG Yaming, et al. Path planning of multi-agent robot based on coevolution [J]. Computer Systems and Applications, 2010, 19(11):157-161. (in Chinese)  
雷小宇, 杨胜跃, 张亚鸣, 等. 基于协同进化的多智能体机器人路径规划 [J]. 计算机系统应用, 2010, 19(11):157-161.
- [54] WU Liang, HE Qinghua, HUANG Zhixiong, et al. Centralized and coordinated path planning of multi robots based on ant colony algorithm [J]. Robot Technology and Application, 2006(3):42-47. (in Chinese)  
吴靓, 何清华, 黄志雄, 等. 基于蚁群算法的多机器人集中协调式路径规划 [J]. 机器人技术与应用, 2006(3):42-47.
- [55] GU Junhua, MENG Huijie, XIA Hongmei, et al. Research on multi robot path planning based on improved ant colony algorithm [J]. Journal of Hebei University of Technology, 2016, 45(5):28-34. (in Chinese)  
顾军华, 孟慧婕, 夏红梅, 等. 基于改进蚁群算法的多机器人路径规划研究 [J]. 河北工业大学学报, 2016, 45(5):28-34.
- [56] POHL I. Heuristic search viewed as path finding in a graph [J]. Artificial Intelligence, 1970, 1(3):193-204.
- [57] GILON D, FELNER A, STERN R. Dynamic potential search—a new bounded suboptimal search [C]//Proceedings of the 9th Annual Symposium on Combinatorial Search. Washington D. C., USA: IEEE Press, 2016:45-65.
- [58] WALKER T T, STURTEVANT N R, FELNER A. Extended increasing cost tree search for non-unit cost domains [C]//Proceedings of International Joint Conference on Artificial Intelligence. Washington D. C., USA: IEEE Press, 2018:534-540.
- [59] PEARL J, KIM J H. Studies in semi-admissible heuristics [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1982, 4(4):392-399.
- [60] BARER M, SHARON G, STERN R, et al. Suboptimal variants of the conflict-based search algorithm for the multi-agent path finding problem [J]. Frontiers in Artificial Intelligence and Applications, 2014, 263:961-962.
- [61] COHEN L, KOENIG S. Bounded suboptimal multi-agent path finding using highways [C]//Proceedings of International Joint Conference on Artificial Intelligence. Washington D. C., USA: IEEE Press, 2016:3978-3979.
- [62] COHEN L, URAS T, KUMAR T K S, et al. Improved solvers for bounded-suboptimal multi-agent path finding [C]//Proceedings of International Joint Conference on Artificial Intelligence. [S.l.]: AAAI Press, 2016:677-690.
- [63] BAE H, KIM G, KIM J, et al. Multi-robot path planning method using reinforcement learning [J]. Applied Sciences, 2019, 9(15):3057-3059.
- [64] SURYNEK P. Unifying search-based and compilation-based approaches to multi-agent path finding through satisfiability modulo theories [C]//Proceedings of the 12th Annual Symposium on Combinatorial Search. Washington D. C., USA: IEEE Press, 2019:15-20.