

# CBS框架下面向复杂地图的低拓展度A\*算法

宣志玮<sup>1</sup>,毛剑琳<sup>1</sup>,张凯翔<sup>2</sup>

(1. 昆明理工大学信息工程与自动化学院, 云南昆明 650500; 2. 昆明理工大学机电工程学院, 云南昆明 650500)

**摘要:** A\*算法是机器人路径规划问题中的重要且常用算法之一,在地形复杂的大型地图中,路径点之间的不可视造成A\*算法需要大规模节点拓展才能找到可行的优化路径,由此导致算法对存储空间的需求剧增和求解效率的降低.对此,本文针对基于冲突搜索(Conflict-Based Search, CBS)框架下的低层路径规划问题,引入三角剖分方法,给出固定障碍处理方法,融合可视性优化获得相邻点可视的优化路径,在此基础上提出分段策略,令具有动态冲突处理能力的A\*算法依相邻可视点进行分段路径规划,最终获得低节点拓展度A\*路径规划算法.通过标准地图数据集的仿真实验表明,在复杂地图下本文提出的算法路径长度为A\*算法的98.1%~102.2%,节点拓展量降低85.4%,算法求解时间减少58.1%.

**关键词:** A\*算法; 路径规划; Delaunay三角剖分; 移动机器人; 冲突消解

**中图分类号:** TP242 **文献标识码:** A **文章编号:** 0372-2112(2022)08-1943-08

**电子学报URL:** <http://www.ejournal.org.cn> **DOI:** 10.12263/DZXB.20210718

## Low-Expansion A\* Algorithm Based on CBS Framework for Complex Map

XUAN Zhi-wei<sup>1</sup>, MAO Jian-lin<sup>1</sup>, ZHANG Kai-xiang<sup>2</sup>

(1. Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan 650500, China;

2. Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming, Yunnan 650500, China)

**Abstract:** A\* algorithm is one of the important and commonly used algorithms in path planning of robots. In large map with complex terrain, large-scale node expansion is needed in A\* algorithm to find a feasible optimization path in the case of invisibility between path points. This leads to a sharp increase in the demand for storage space and a decrease in running efficiency of the algorithm. To solve low-level path planning problems based on CBS(Conflict-Based Search) framework, an A\* path planning algorithm with low node expansion is proposed. Triangulation method and fixed obstacle processing method are introduced to obtain an optimized path that is visible to adjacent points. Moreover, a segmentation strategy is proposed based on integrated visibility optimization to plan the segmented path according to the adjacent visible points with dynamic collision processing ability. Simulation experiments on standard map data sets under complex terrain show that the path length of the proposed algorithm in this paper is 98.1%~102.2% as long as that of A\* algorithm, compared with which, the amount of node expansion reduces by 85.4% and the running time of algorithm reduces by 58.1%.

**Key words:** A\* algorithm; path planning; Delaunay triangulation; mobile robot; conflict resolution

### 1 引言

随着移动机器人的应用范围越来越广泛,移动机器人的应用环境也变得越来越复杂、多变,因此面向复杂地图的移动机器人路径规划问题成为机器人领域<sup>[1]</sup>、视频游戏领域<sup>[2]</sup>、无人机领域<sup>[3]</sup>、机器人编队系统领域<sup>[4]</sup>的研究热点.目前针对机器人路径规划的求解算

法有基于A\*(A Star)的算法<sup>[5]</sup>,基于约简(Reduction-Based)的算法<sup>[6]</sup>,也有基于搜索框架的算法<sup>[7,8]</sup>,基于采样的算法<sup>[9,10]</sup>,人工势场法<sup>[11]</sup>,蚁群算法<sup>[12,13]</sup>等.

基于冲突的搜索算法(Conflict-Based Search, CBS)是求解多机器人路径规划(Multi-Agent Path Finding, MAPF)的一种重要框架<sup>[7]</sup>,由两层结构组成,低层结构

采用 A\* 算法完成单机器人路径规划; 高层结构进行多机器人路径的冲突判断, 并向低层提供冲突点和冲突时间信息. 常用 A\* 类算法作为低层寻路算法求解路径, 为了提高算法求解效率, 其中有 Tie-Breaking 策略<sup>[14,15]</sup>, 通过减少搜索过程中拓展节点的数量来提升运行效率. A\* 算法处理大规模任务时仍然会拓展大量节点, 这在计算上是不可行的<sup>[16]</sup>.

针对低层算法在面对大地图和复杂障碍物时求解效率低的问题, 本文提出带障碍处理的 Delaunay 三角剖分优化和分段 A\* 算法相结合的思想, 对 CBS 框架下低层路径规划问题进行求解, 以此完成复杂障碍环境中高效地路径优化.

## 2 问题描述

### 2.1 Conflict-Based Search(CBS)框架描述

A\* 算法在求解 MAPF 问题时, 状态空间与机器人数量呈指数关系<sup>[6]</sup>, 当机器人数量为 1 时, 状态空间只与地图大小成线性关系. CBS 框架将 MAPF 问题分解为一个带约束限制的单机机器人寻路问题. CBS 框架定义如下所示:

给定地图  $G=(V, E)$ ,  $V$  为图中的顶点集合,  $E$  为图中的边集合. 约束元组信息  $(a_i, v, t)$  表示机器人  $a_i$  禁止在时刻  $t$  占据顶点  $v, v \in V$ . 每个机器人最后生成的路径都满足对其添加的约束.

CBS 的核心思想是为冲突机器人增加一组约束, 直到找到能满足约束的路径. CBS 算法是两层结构算法. 在高层, 进行全局冲突检测并添加约束; 低层则更新机器人路径以满足新的约束信息. 如图 1 所示.

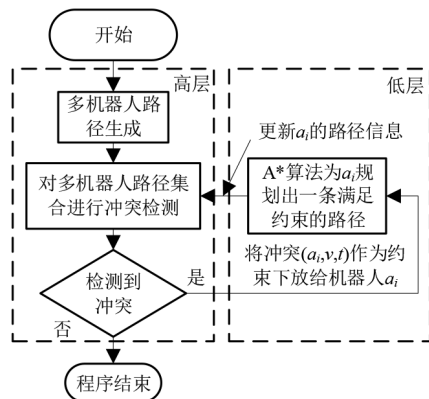


图 1 CBS 算法对冲突处理的两层结构

### 2.2 低层机器人路径规划问题描述

机器人的最优路径规划问题为依据某个或某些优化准则(如工作代价最小、行走路径最短、行走时间最短等), 在给定工作空间中找到一个从起始点到目标点的无障碍最优路径.

给定栅格地图  $G$  和机器人  $a$ , 机器人起点为  $s$  和终点为  $g$ , 经过时间为  $T$ , 时间离散化, 机器人每次移动时间为一个单位时间, 允许向四邻域范围内的自由栅格进行节点拓展, 最后即可生成最终路径  $P$  为一组路径序列  $P=(s^0, p^1, \dots, p^{T-1}, g^T)$ , 注意, 该路径集合  $P$  为无固定障碍路径.

在 CBS 框架下, 低层求解器采用考虑时空信息的标准 A\* 算法进行求解. 标准 A\* 算法在地形简单、范围较小的地图中寻路求解路径质量高, 且在该框架下, A\* 算法仅与地图规模是线性关系, 但随着机器人数量的增加, 算法仍然会拓展数量众多的节点, 这对求解效率的提升带来了阻碍.

本文通过计算几何和优化启发式函数选择节点策略, 两者相结合提高 CBS 框架下低层算法的求解效率, 加速 MAPF 的求解过程.

## 3 带障碍处理 Delaunay 三角剖分优化的低拓展度 A\* 算法

为降低 A\* 算法所需的节点扩展度从而提升求解效率, 本文提出带障碍处理 Delaunay 三角剖分优化的低拓展度 A\* 算法 (Low Extension A\* algorithm under Delaunay Triangulation with Obstacle Constraints, LEADTOC), 采用 Delaunay 三角剖分进行地图和固定障碍物的处理<sup>[17]</sup>, 然后采用最短路径算法和可视化方法获得无固定障碍的连通路径, 最后采用分段 A\* 算法规划出优化路径.

### 3.1 带障碍处理的 Delaunay 三角剖分及无固定障碍路径优化

Delaunay 三角剖分是指把散点集合  $C$  剖分成不均匀的三角形网格. 对于平面内的给定散点有以下 2 个特点: 空圆性 (在 Delaunay 三角剖分中任意三角形的外接圆范围内不会有其他点存在), 该特性保证了 Delaunay 三角形的唯一性; 最大化最小角特性 (剖分后三角形的最小角最大).

通过缩放因子的灵活选择可控制生成三角剖分地图边和节点的数量, 对特定地图选取恰当的缩放因子可在保证求解质量的前提下, 进一步优化求解效率, 缩放因子一般取为 5~8. 以  $250 \times 250$  大小的地图为例, 取缩放因子为 7, 可将大地图近似缩放为  $35 \times 35$  大小规模的地图.

本文地图建模时采用栅格法建模. 每个栅格与相邻栅格的距离为 1.

对于给定地图, 首先进行剖分散点的选择, 选出两部分的点: 包围障碍物边缘的点和二维平面内的均匀取点. 包围障碍物边缘的散点, 这一部分的散点距离最近障碍物的距离为 1, 即该部分选取的每一个散点与其

最近的障碍物的距离均为 1, 如图 2 所示; 第二部分散点根据缩放规模在二维栅格地图均匀取点, 点间距为缩放因子所确定的缩放规模, 缩放规模为 5 就是在横纵坐标两个维度每隔 5 个栅格取一个点 (不包括处于障碍物内的点), 剖分散点由以上两部分点构成。

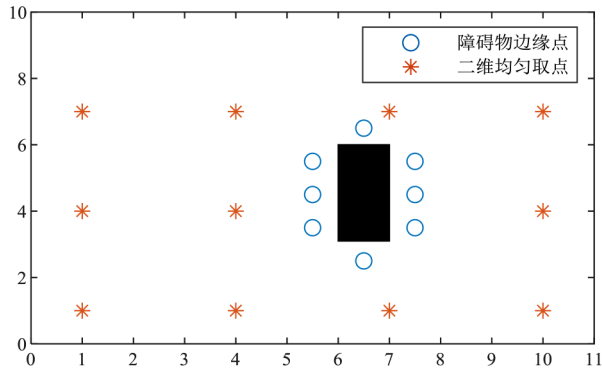


图 2 剖分散点选择

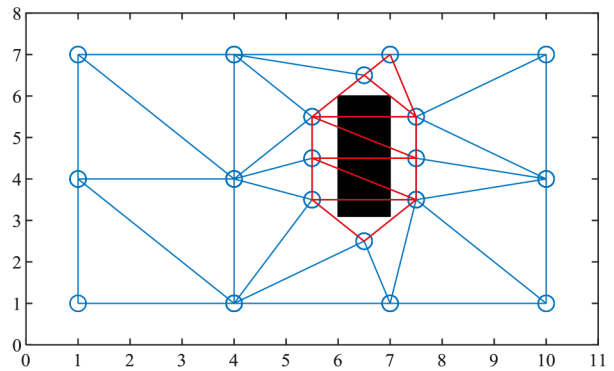
如图 3 所示, 根据散点生成的第一次三角形, 其中部分三角形的边有穿越障碍物的情况. 通过最近点搜索算法, 将三角形的重心设置为参考点, 障碍物点作为查询点, 筛选出包围查询点的三角形. 后将筛选出的三角形的每两个顶点间进行线性插值, 插值后的状态点有一个及其以上的点处于障碍物栅格则视为这两个顶点之间的边不可见, 将三角形组成的不可见边进行删除, 最后生成基于三角剖分的无障碍连通图  $G_D$  (Graph with Constrained Delaunay Triangulation).

随后, 对无障碍连通图  $G_D$  采用最短路径算法计算从起点  $s$  到终点  $g$  的最短路径, 可获得机器人  $a$  在图  $G_D$  上的路径集合  $P=(s, v_1, \dots, v_n, g), v_i(i=1, \dots, n)$  为中间路径点,  $s, g, v_i \in V$ .

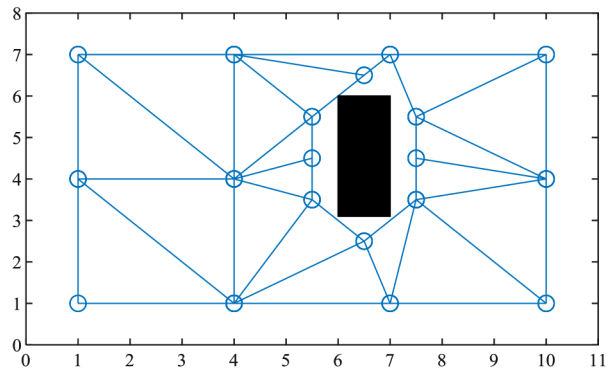
根据两点间直线距离最短原理, 采用可视性判断对路径集合  $P$  从起点  $s$  到终点  $g$  进行优化, 将当前节点与序号大于当前节点的最大序号可视节点连接, 连接后将当前序号最大的可视节点视为起点与其后续节点继续做可视性分析, 最终得到优化后的路径集合. 如图 4 所示, 通过最短路径算法得到路径集合  $P=(s, v_1, v_2, v_3, v_4, g)$ , 经过可视性优化后, 得到路径集合  $P_{opt}=(s, v_4, g)$ . 第一次生成的路径点集合包含冗余节点, 通过可视性原则对路径集合  $P$  进行优化, 删除冗余节点, 得到二次筛选节点集合  $P_{opt}$ , 低层寻路算法按照二次筛选路径点集合  $P_{opt}$  进行分阶段寻路。

### 3.2 基于可视点的 A\* 算法

在可视优化路径的基础上, 采用 A\* 算法依据可视点进行分段寻路, 即对于机器人  $a$  由初始任务  $\text{Task}=(s, g)$  替换为分阶段寻路任务  $\text{Task}_{opt}=(P_{opt})$ . 需说明的是, 在每个阶段, 机器人在栅格地图采用 A\* 算法进行四



(a) 未考虑障碍物约束



(b) 考虑障碍物约束

图 3 带障碍物处理的三角剖分

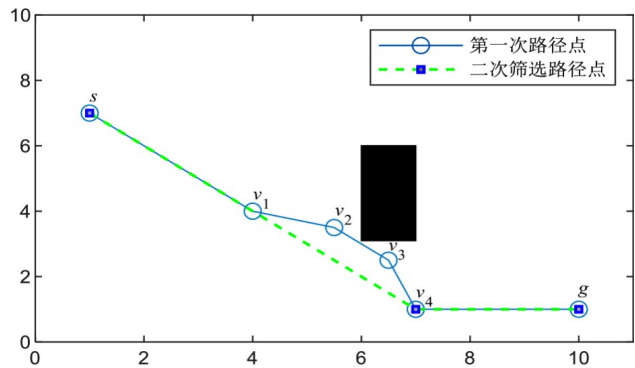


图 4 剖分散点二次筛选

邻域搜索, 即给定起点的情况下, 向邻近无障碍栅格搜索, 如图 5 所示, 机器人可以向上、左、右三个直线方向上的邻近空白区域拓展节点, 但无法向下拓展节点, 由此, 避开障碍物。

综上所述, 整个改进算法如图 6 所示, 由地图处理、分阶段寻路、冲突检测三部分组成. 在带约束的三角剖分地图基础上, 进行图搜索和经过点的可视性优化, 然后进行分阶段 A\* 算法寻路. 在 CBS 算法框架下, 算法不仅能快速找到最优路径, 同时能够满足约束条件, 最后生成一条无冲突的路径。

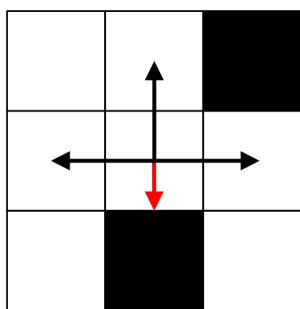


图5 机器人节点拓展

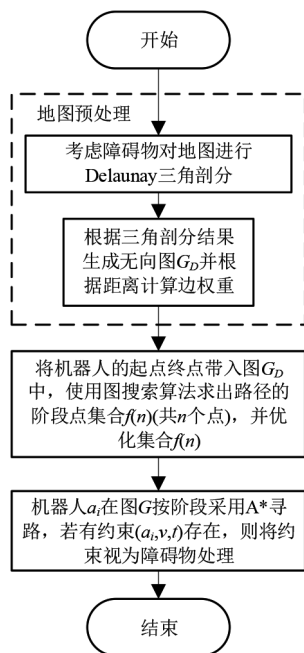


图6 LEADTOC算法流程

### 4 仿真结果

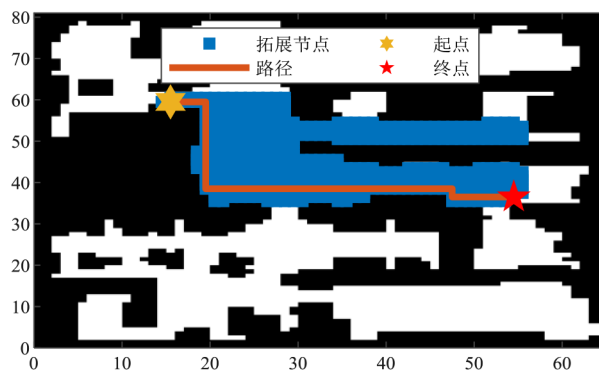
为测试所提出的 LEADTOC 算法性能, 本文采用文献[18]中给出的标准算例集, 该算例集给出不同规模和复杂度的地图, 以及机器人在地图中的起点和终点, 由此进行的测试具有可复现性和可对比性. 对比算法为考虑时空信息的标准 A\* 算法<sup>[5]</sup>和在扩展度方面有优势的 Tie-Breaking A\* 算法<sup>[14]</sup>, 在本文中 Tie-Breaking 策略取为:  $h = (1 + d) \times h$ , 其中,  $h$  是启发式函数, 为曼哈顿距离,  $d$  为 Tie-Breaking 值, 是一个很小的数字, 本文 A\* 算法、Tie-Breaking A\* 算法和 LEADTOC 算法的  $d$  值分别取为 0、0.001、0.001. 所有算例结果均为 20 次独立运行的平均值. 测试采用的性能参数为: 扩展节点数量, 最终路径长度, 求解时间(完成每次规划所需的时间). 所有运行均在 matlab2020b 环境下, 11th Gen Intel (R) Core (TM) i5-1135G7 @ 2.40 GHz 和 16 GB 内存容量的计算机配置下完成.

### 4.1 算法求解效率对比

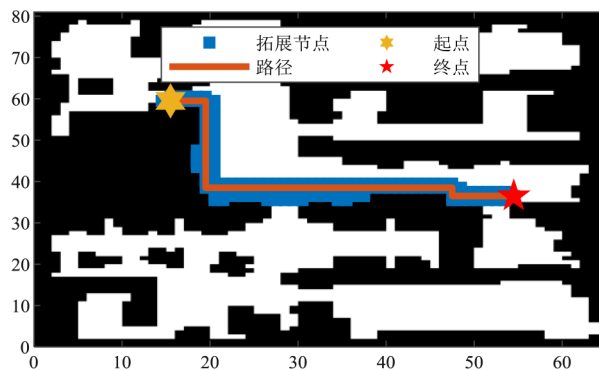
#### 4.1.1 简单测试算例

本算例为地图 den312d 下的测试, 机器人起点坐标为 (16, 60), 终点坐标为 (55, 37). 3 种算法的路径规划结果如图 7 所示, 其中红色线为最终规划的路径, 蓝色区域为算法运行过程中扩展的节点. 由图 7 可见, 本文提出的 LEADTOC 算法和 Tie-Breaking A\* 算法扩展节点数量都明显少于 A\* 算法. 3 种算法的性能结果见表 1.

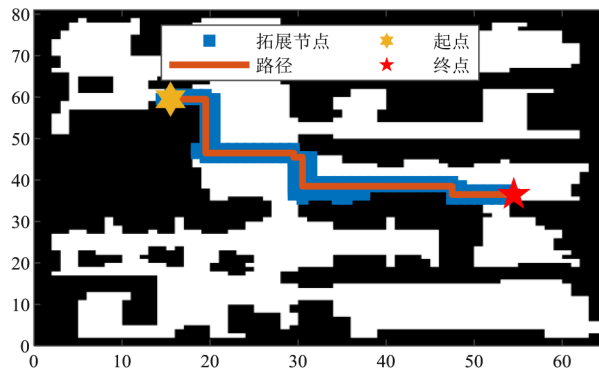
由表 1 可见, 3 种算法规划的路径长度相同. 在求解时间方面, 本文算法高于 Tie-Breaking A\* 算法. 这是



(a) A\*算法



(b) Tie-Breaking A\*算法



(c) LEADTOC算法

图7 简单算例3种算法节点拓展数量对比

表 1 简单算例仿真结果

算法	运行时间/ms	节点拓展数量	路径长度
A*算法	40.98	598	62
Tie-Breaking A*算法	8.91	181	62
LEADTOC算法	16.13	178	62

由于本文方法采用了三角剖分进行地图预处理和无障碍连通图规划, 导致计算时间稍长于 Tie-Breaking A\* 算法. 但经过预处理后依然可以通过降低节点扩展量来减少求解时间, 所以求解时间仍能比 A\* 算法低.

#### 4.1.2 复杂测试算例

图 8 给出了复杂地图 den520d 下的路径规划结果,

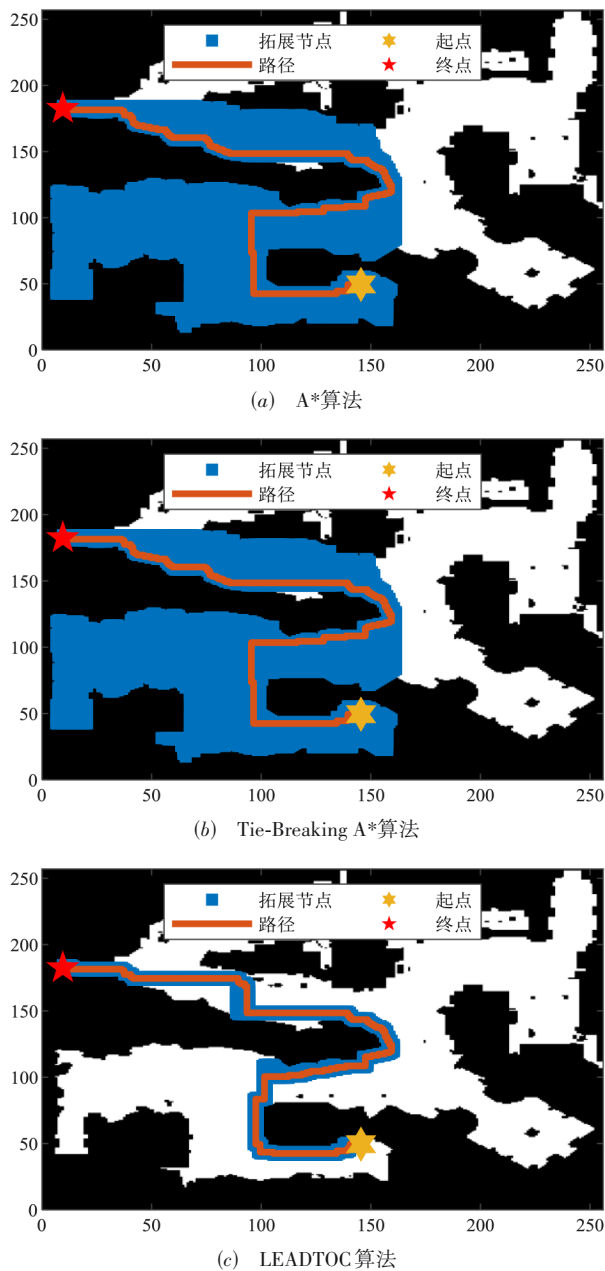


图 8 复杂算例 3 种算法节点拓展数量对比

其中, 机器人起点坐标为 (146, 50), 终点坐标为 (10, 183), 起点和终点间的障碍物较为曲折复杂. 表 2 则给出各算法的性能指标结果.

由测试结果可见, 在复杂地图上本文算法均可在拓展节点数量和计算时间上明显少于 A\* 算法和 Tie-Breaking A\* 算法. 这说明本文算法采用的三角剖分优化虽然在地图预处理和无障碍连通图的生成方面有时间代价, 但是当地图场景较为复杂时, 该时间代价远低于节点扩展带来的时间消耗, 故可较大提升算法的栅格路径规划效率.

表 2 复杂算例仿真结果

算法	运行时间/ms	节点拓展数量	路径长度
A*算法	610.46	12 763	415
Tie-Breaking A*算法	616.72	12 763	415
LEADTOC算法	99.97	1 265	411

#### 4.2 更多测试算例

为了进一步验证本文算法的求解效率和求解质量, 与 A\* 算法和 Tie-Breaking A\* 算法对比, 对文献 [18] 中的 10 个地图进行测试. 图 9 给出了若干典型地图样貌, 由图可见不同复杂程度和复杂类型的障碍情况. 表 3 给出每个测试地图可用算例数量, 可用算例数量指在 MATLAB 笛卡尔坐标系下的可用算例数量, 表 4 给出了 10 个算例下的算法性能结果.

由表 4 可得, LEADTOC 算法在节点拓展数量上为 A\* 算法的 5.3%~52.2%, 路径长度为 A\* 算法的 98.1%~102.2%, 平均求解时间为 A\* 算法的 44.3%.

LEADTOC 算法在不规则复杂地形表现良好, 在规则地形如 Room 地形的求解质量稍差, LEADTOC 算法在大地形、复杂地形能发挥出改进算法的求解效率的优势.

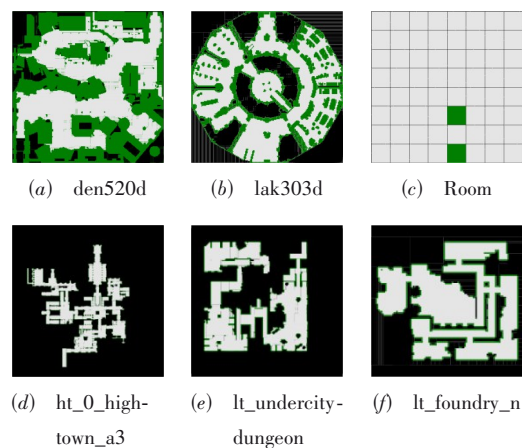


图 9 测试地图集

表 3 测试地图可用算例数量

地图名称	地图大小	算例数量
den520d	257×256	269
den312d	81×65	121
lak303d	194×194	474
ht_0_hightown_a3	514×514	179
lt_undercitydungeon	177×184	312
lt_foundry_n	92×109	202
8room_000	512×512	1 607
16room_000	512×512	1 839
32room_000	512×512	1 658
darkforest	512×512	494

表 4 Benchmark 测试结果

地图	平均拓展节点数			平均路径长度			平均运行时间/ms		
	A*	Tie-Breaking A*	LEADTOC	A*	Tie-Breaking A*	LEADTOC	A*	Tie-Breaking A*	LEADTOC
den520d	3 574	3 062	540	174.78	175.25	174.17	155.43	131.20	37.31
den312d	447	321	168	60.36	60.86	60.35	18.80	13.25	13.58
lak303d	4 126	4 021	575	200.77	205.32	201.36	194.99	189.89	137.75
ht_0_hightown_a3	3 717	3 398	538	184.65	185.80	184.78	170.01	160.37	77.46
lt_undercitydungeon	2 873	2 706	442	165.44	165.97	165.37	129.56	122.25	36.39
lt_foundry_n	1 122	1 078	245	91.34	91.66	91.42	49.75	47.41	20.24
8room_000	22 839	21 981	1 344	491.60	494.01	498.54	1 015.55	983.94	517.62
16room_000	25 376	24 522	1 515	495.08	496.56	502.60	1 126.83	1 084.69	378.58
32room_000	26 877	25 844	1 362	436.28	436.81	446.75	1 209.36	1 153.37	297.31
darkforest	8 797	5 680	965	290.55	291.17	293.70	390.25	242.47	110.12

下面是上述算例的结果分析说明:如表 5 所示,以 A\*算法为基准,基准设置为 1,分别对比了 A\*比 A\*、Tie-Breaking A\*比 A\*、LEADTOC 比 A\*在节点拓展数量、路径长度、运行时间三方面的算法性能对比情况.由结果可得 LEADTOC 算法在节点数量拓展对比,运行时间对比均低于另外 2 种算法,在路径长度对比方面,长度基本持平.

表 5 算法性能比较

算法	运行时间比	节点拓展数量比	路径长度比
A*算法	1	1	1
Tie-Breaking A*算法	0.887	0.890	1.006
LEADTOC 算法	0.419	0.146	1.006

### 4.3 动态冲突场景仿真测试

本节仿真实验地图为 Benchmark 中的 den520d,机器人和动态冲突的参数设置见表 6.

如图 10(a)所示,在不考虑动态冲突时,机器人在规划路径上将分别与动态冲突物 1 和 2 在冲突点相撞.对此,本文算法将冲突碰撞信息转化为约束传递给机器人,由机器人规划出一条无冲突路径,行进过程中可

表 6 仿真场景参数设置表

属性	起点	终点
机器人	(16,60)	(55,37)
动态冲突物 1	(20,46)	(20,62)
动态冲突物 2	(61,21)	(22,39)

安全避开 2 个动态冲突,成功到达终点,如图 10(b)所示.可见,本文算法能很好地处理动态冲突物的碰撞信息,并规划出机器人的无冲突路径.

## 5 结论

低层路径规划质量是影响多机器人路径规划的重要因素之一,对此,本文提出 LEADTOC 算法,其中引入带障碍处理的三角剖分方法,并提出缩放因子以灵活地适应不同类型的地图且同时降低地图数据量,由此获得无固定障碍的路径引导.进一步采用可视化分段策略,令具有动态冲突处理能力的 A\*算法依相邻视点进行分段路径规划,该策略可大大降低 A\*算法在路径探索时的节点拓展度.仿真结果表明,复杂地形下,LEADTOC 算法的节点拓展数量和存储空间需求量较

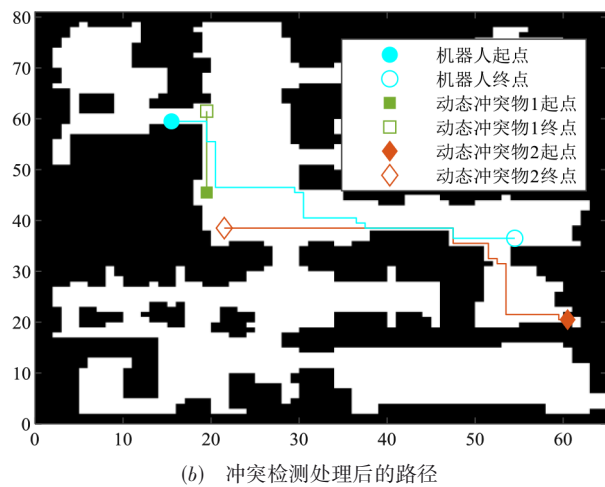
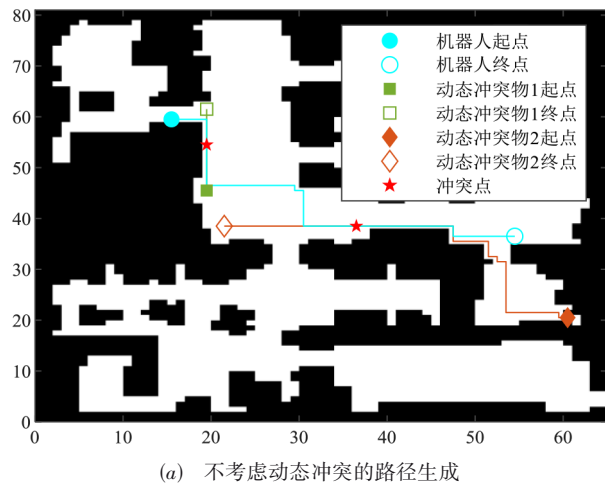


图 10 动态冲突物仿真结果

低,在不降低求解质量的情况下能较好地提升求解效率.后续将对 CBS 框架下多机器人的冲突判断和分解策略的优化展开进一步研究.

#### 参考文献

- [1] MA H, LI J Y, KUMAR T K S, et al. Lifelong multi-agent path finding for online pickup and delivery tasks[C]//AAMAS' 17: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems. São Paulo, Brazil: IFAAMAS Press, 2017: 837-845.
- [2] LI J Y, Sun K X, MA H, et al. Moving agents in formation in congested environments[C]//AAMAS' 20: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. Auckland, New Zealand: IFAAMAS Press, 2020: 726-734.
- [3] HO F, SALTA A, GERALDES R, et al. Multi-agent path finding for UAV traffic management[C]//AAMAS' 19: Proceedings of the 18th International Conference on Autono-

mous Agents and MultiAgent Systems. Montreal, Canada: IFAAMAS Press, 2019: 131-139.

- [4] 王保防,张瑞雷,郭健,等.纵向打滑状态下的轮式移动机器人编队控制[J].电子学报,2017,45(1):206-212.  
WANG B F, ZHANG R L, GUO J, et al. Formation control for car-like mobile robots under slip conditions[J]. Acta Electronica Sinica, 2017, 45(1): 206-212. (in Chinese)
- [5] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [6] SURYNEK P, FELNER A, STERN R, et al. Efficient SAT approach to multi-agent path finding under the sum of costs objective[C]//ECAI' 16: Proceedings of the Twenty-second European Conference on Artificial Intelligence. The Hague, Netherlands: IOS Press, 2016: 810-818.
- [7] SHARON G, STERN R, FELNER A, et al. Conflict-based search for optimal multi-agent pathfinding[J]. Artificial Intelligence, 2015, 219: 40-66.
- [8] 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.
- [9] 王乐乐, 睦泽智, 蒲志强, 等. 一种改进 RRT 的多机器人编队路径规划算法[J]. 电子学报, 2020, 48(11): 2138-2145.  
WANG L L, SUI Z Z, PU Z Q, et al. An improved RRT algorithm for multi-robot formation path planning[J]. Acta Electronica Sinica, 2020, 48(11): 2138-2145. (in Chinese)
- [10] 尹高扬, 周绍磊, 吴青坡. 基于改进 RRT 算法的无人机航迹规划[J]. 电子学报, 2017, 45(7): 1764-1769.  
YIN G Y, ZHOU S L, WU Q P. An improved RRT algorithm for UAV path planning[J]. Acta Electronica Sinica, 2017, 45(7): 1764-1769. (in Chinese)
- [11] 马小陆, 梅宏. 基于改进势场蚁群算法的机器人全局路径规划[J]. 机械工程学报, 2021, 57(1): 19-27.  
MA X L, MEI H. Mobile robot global path planning based on improved ant colony system algorithm with potential field[J]. Journal of Mechanical Engineering, 2021, 57(1): 19-27. (in Chinese)
- [12] DORIGO M, BIRATTARI M, STUTZLE T. Ant colony optimization[J]. IEEE computational intelligence magazine, 2006, 1(4): 28-39.
- [13] 许凯波, 鲁海燕, 黄洋, 等. 基于双层蚁群算法和动态环境的机器人路径规划方法[J]. 电子学报, 2019, 47(10): 2166-2176.  
XU K B, LU H Y, HUANG Y, et al. Robot path planning

- based on double-layer ant colony optimization algorithm and dynamic environment[J]. Acta Electronica Sinica, 2019, 47(10): 2166-2176. (in Chinese)
- [14] CORRÊA A B, PEREIRA A G, RITT M. Analyzing Tie-Breaking strategies for the A\* algorithm[C]//IJCAI18' 19: Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm, Sweden: AAAI Press, 2018: 4715-4721.
- [15] GOLDENBERG M, FELNER A, STERN R, et al. Enhanced partial expansion A[J]. Journal of Artificial Intelligence Research, 2014, 50: 141-187.
- [16] FELNER A, STERN R, SHIMONY S E, et al. Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges[C]//Proceedings of the Tenth International Symposium on Combinatorial Search. Pittsburgh, Pennsylvania, USA: AAAI Press, 2017: 29-37.
- [17] CHEW L P. Constrained delaunay triangulations[J]. Algorithmica, 1989, 4(1): 97-108.
- [18] STURTEVANT N R. Benchmarks for grid-based pathfinding[J]. IEEE Transactions on Computational Intelligence and AI in Games, 2012, 4(2): 144-148.

#### 作者简介



宣志玮 男, 1995年12月出生, 云南蒙自人. 昆明理工大学信息工程与自动学院硕士研究生. 主要研究方向为机器人路径规划.  
E-mail: rangxuan@foxmail.com



毛剑琳(通讯作者) 女, 1976年5月出生, 广西桂林人. 昆明理工大学教授, 博士生导师. 主要研究方向为通信网络资源分配与协议优化、网络化控制系统研究、智能优化及调度算法研究等.  
E-mail: 1318524654@qq.com



张凯翔 男, 1993年10月出生, 浙江台州人. 昆明理工大学机电工程学院博士研究生. 主要研究方向为机器人路径规划.  
E-mail: zhangkaixiang@stu.kust.edu.cn