文章编号: 1001-0920(2017)08-1395-08 DOI: 10.13195/j.kzyjc.2016.1134

# 基于方向约束的A*算法

李 冲†, 张 安, 毕文豪

(西北工业大学 电子信息学院，西安 710072)

**摘 要:** 实际机器人路径规划问题经常需要考虑路径的转弯约束以及路径起始/目标角要求,为此提出一种基于方向约束的A*算法. 新算法区分同一路径点处不同方向的各条路径,通过定向扩展机制来满足路径方向约束,并采用节点合并策略和不一致队列降低算法复杂度. 理论分析和典型地图集上的实验结果证明,所提算法总是能够保证给出符合转弯约束和起始/目标角约束的最短路径,且相比于现有算法,能够有效提高方向约束路径规划问题的求解能力.

**关键词:** 路径搜索；A*算法；转弯约束；角度约束

**中图分类号:** TP18 **文献标志码:** A

# A* algorithm based on direction constraints

*LI Chong†, ZHANG An, BI Wen-hao*

(School of Electronics and Information，Northwestern Polytechnical University，Xi'an 710072，China)

**Abstract:** The turning constraint and start/goal angle constraint are often needed to be considered in the actual robot path planning problem, and the A* algorithm based on direction constraints is proposed in this paper. Paths in the same path point are distinguished by their different directions in the algorithm, and the direction constraints of paths are met by the directional extension mechanism, while the node merging strategy and the inconsistent list are adopted to reduce the algorithm complexity. Theoretical analysis and experimental results on the typical map set show that the propoved algorithm can always guarantee to give the shortest path which conforms to the turning constraint and start/goal angle constraint. Compared with the existing algorithms, the proposed algorithm can improve the ability of solving path planning problem with direction constraints effectively.

**Keywords:** path search；A* algorithm；turning constraint；angle constraint

## 0 引 言

路径规划是机器人、人工智能等领域的一个重要的基本问题. 目前的研究热点主要集中在缩短规划时间、 减小路径长度和降低预处理代价3个方面. 通常,移动对象(例如机器人)被简化为有一定体积、可任意自由平移的单位,一条可行路径仅要求移动对象不与障碍物发生碰撞. 然而,实际机器人运动系统中普遍存在非完整性约束[1],即使在无障碍区域,机器人也只能在一定转弯角范围内作受限运动,而不允许大角度转折或平移. 因此,传统自由位形空间内的无碰撞路径不一定是实际工作空间内的可行路径. 另外,在诸如进/出泊车位、定向拍照等任务中,路径的起始角或目标进入角必须满足特定的要求[2],现有的基于自由位形空间的算法并不能很好地处理这些约束. 文献[3-4]采用有约束网络图环境模型,将路径的转弯约束和角度约束转化为特定网络节点的连通性约束,从而解决交通网络中的有约束路径规划问题,但网络图建模的限制很多,仅适用于城市街道和公路网等特定环境. 文献[5-7]采用几何方法,分别基于Bezier曲线、Clothoid曲线和Dubins曲线,规划出由曲线和其切线组成的平滑路径,通过曲线切入角和曲率半径实现路径的方向约束. 文献[8-11]则采用了导引律控制的思想,在从起始点逐渐接近目标的过程中,监测当前位置与目标间的视线角,根据相应导引律调整当前运动方向,以实现指定末端进入角,并通过法向加速度阈值约束转弯半径. 几何方法和导引律控制方法只能处理无障碍物环境,目前仅应用于高空飞行器或制导武器. 文献[12-13]在规划过程

**收稿日期:** 2016-09-06；**修回日期:** 2017-02-12.

**基金项目:** 国家自然科学基金项目(61573283).

**作者简介:** 李冲(1989−), 男, 博士, 从事自主控制和人工智能的研究；张安(1962−), 男, 教授, 博士生导师, 从事智能指挥与控制工程等研究.

†通讯作者. E-mail: lichong@mail.nwpu.edu.cn

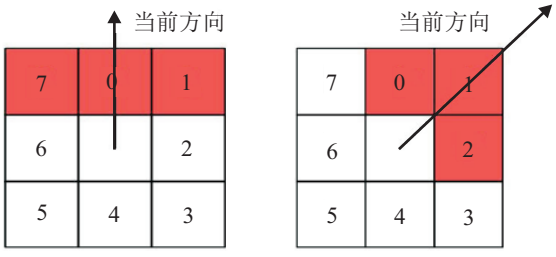
中设置临时障碍或者运动学约束禁止机器人非法转向,以限制路径转弯,并通过设置虚拟障碍区来约束起始/目标角,这些方法会损失可行解空间,影响算法完备性. 文献[14]研究了在起始/目标角方向设置临时起始/目标的方法,但这会导致路径始/末端方向不可控,容易造成大角度转折.

本文提出一种基于方向约束的A*算法. 该算法在自由位形空间基础上引入姿态维度,用于区分同一路径点处不同方向的各条路径,以保证解空间的完备性,并通过定向扩展机制满足路径的转弯约束和起始/目标角要求. 理论分析和实验对比结果表明,相比于现有算法,新算法总是能够保证给出符合转弯约束和起始/目标角要求的最短路径.

## 1 路径规划的方向约束问题

栅格化是目前最常用的环境建模方法之一[15]. 本文采用八向栅格图,考虑到机器人实际碰撞体积不可能为零,为安全起见,机器人斜向移动时,需要保证与运动方向垂直的两个相邻栅格均为可通行栅格.

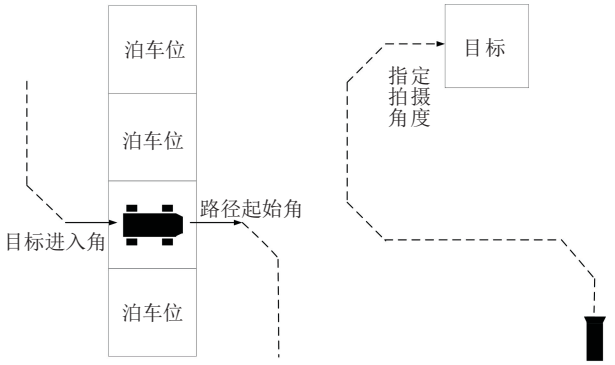
机器人路径转弯约束是指连续两次移动之间,机器人运动方向变化夹角不能超过一定范围,其具体值与路径单步步长和机器人转弯能力有关. 栅格图中,机器人相对于栅格的转弯能力可以用转弯等级表示. 机器人每次向下一步移动时,共有8个可选方向,将其编号如图1所示. 转弯等级$T$表示机器人下一步运动方向与当前方向间的最大允许偏移程度. 显然,机器人每次运动共有$2T+1$个合法方向. 图1中机器人位于中央栅格,长箭头为当前运动方向,深色部分为$T=1$时的合法下一步栅格.




(a) 当前运动方向0 (b) 当前运动方向1

**图1 转弯能力等级$T=1$时的运动约束**

路径方向约束的另一种情况是起始/目标角约束. 以图2(a)为例,为保证在泊车位中停留时的正确姿态,机器人必须从特定角度进入或离开泊车位;而图2(b)中,任务要求机器人必须拍摄目标左侧的照片.




(a) 进入泊车位 (b) 定位拍照

**图2 起始/目标角约束问题**

## 2 A*算法

A*算法近年来被广泛应用于路径规划领域[16]. 由于具有最优性、完备性和高效性,其系列改进算法一直是路径规划领域的研究热点[17-18]. 该算法是一种启发式算法,其启发式代价为

$$p.\mathrm{fval}=p.\mathrm{gval}+p.\mathrm{hval}. \tag{1}$$

其中:gval为起始点$S$到栅格点$p$的当前路径长度,hval为栅格点$p$与目标点$G$间的估计代价,fval为栅格点$p$的总启发式代价.

如果某一父节点可以使得子节点的gval值减小,则称子节点被父节点更新. A*算法总是选取具有最小总启发式代价fval的待扩展节点,将其移入已扩展队列Close_list,并将被其更新的子节点移入待扩展队列Open_list. 如果选取的估计代价hval值总是不超过栅格点与目标点间的实际路径长度,则A*算法保证获得最优路径. 八向栅格图中,记相邻栅格点间的横向或竖向移动距离为单位长度1,斜向移动距离为1.414,估计代价hval可以采用octile距离,计算方法为

$$\begin{aligned} &p.\mathrm{hval}=\mathrm{octile}(p,G)=\\ &1.414\times\min(\mathrm{d}x,\mathrm{d}y)+|\mathrm{d}x-\mathrm{d}y|. \end{aligned} \tag{2}$$

其中:$dx=|x_G-x_p|$,$dy=|y_G-y_p|$,$x_p$、$y_p$、$x_G$和$y_G$分别是栅格点$p$和目标点$G$的横纵坐标.

显然,现有A*算法并没有考虑机器人在各个路径点处的运动方向,也就不能约束路径的转弯程度和起始/目标角. 文献[12-13]通过设置临时障碍或者运动学约束禁止机器人非法转向,以限制路径转弯. 但是,由于A*算法只保留当前长度最短的唯一一条路径,并舍去来自其他方向的路径信息,上述禁忌限制总是会损失可行解空间. 图3中,A*算法只为栅格点$(3,2)$保留实线所示的“最短”路径,机器人在该栅格处运动方向向上,当转弯能力$T=1$时,A*算法无法找到可行解.

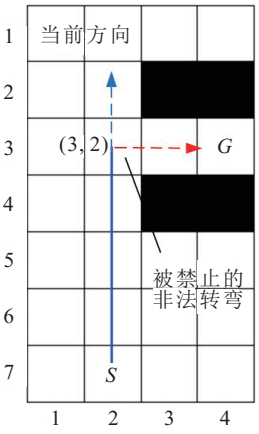
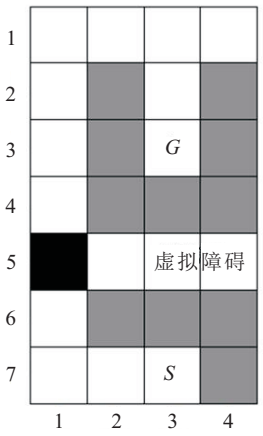


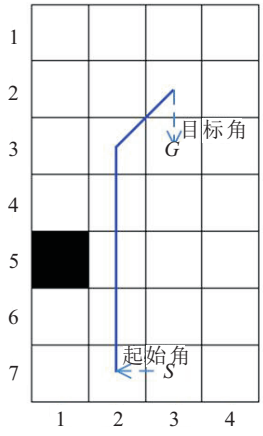

**图 3**　A*算法禁止非法转弯

现有A*算法对起始/目标角约束的处理方法包括[13-14]: 设置虚拟障碍区, 以约束起始/目标角; 在起始/目标角方向设置临时起始/目标, 规划结束后再接入真实起始/目标点. 但是, 虚拟障碍区会造成可行解空间的损失, 影响算法完备性, 且无法处理45°斜向约束角, 而采用临时起始/目标点时, 路径末端方向不可控, 极易产生大角度转折, 导致路径不可执行.

图4中, $S$ 为起始点, $G$ 为目标点, 任务要求机器人起始运动方向向左, 并从目标上方抵达终点. 假设机器人转弯能力为 $T=2$ 级, 黑色为障碍栅格. 图4(a)采用虚拟障碍区法, 虚拟障碍(灰色栅格)导致无可行解, 图4(b)中栅格(2, 3)为临时目标点, 真实目标点 $G$ 接入路径后, 出现135°转折, 超出机器人转弯能力.




(a)　虚拟障碍法




(b)　临时起始/目标点法

**图 4**　现有起始/目标角约束处理方法

## 3　基于方向约束的A*算法

为解决上述问题, 本文提出一种基于方向约束的A*算法, 其基本思路是在A*算法的基础上引入姿态维度. 对于同一栅格点, 区分和存储从不同角度抵达该点的各条路径, 并通过定向扩展机制满足路径的转弯约束和起始/目标角要求.

### 3.1　算法原理

采用一个与地图行列数相同的矩阵代表地图环境, 矩阵的每一个元素对应一个栅格路径点.

**定义 1**　基于方向约束的A*算法中, 一个搜索节点 $p$ 由一组属性值组成: {gval[8], hval, fval}. 其中: $p.\mathrm{gval}[i]$ 为抵达路径点 $p$ 时方向为 $i$ 的路径长度, $i=0,1,\cdots,7$; $p.\mathrm{hval}=\mathrm{octile}(p,G)$ 为路径点 $p$ 与目标点 $G$ 间的估计代价; $p$ 点的总启发式代价取各方向路径启发式代价的最小值, 即

$$p.\mathrm{fval}=\min\{p.\mathrm{gval}[i]|i=0,1,\cdots,7\}+p.\mathrm{hval}. \tag{3}$$

显然, 引入姿态维度后, 同一栅格点处包含8个代表不同方向的路径, 且彼此相互独立. 但通过将同一路径点处的不同方向路径合并成为一个搜索节点, 并共享hval, 栅格点与搜索节点仍然一一对应. 另外, 各条路径方向明确, 自动对应唯一的父代路径点, 无需额外记录. 例如, $p.\mathrm{gval}[0]$ 表示抵达栅格点 $p$ 时机器人速度方向向上的那条路径的长度, 显然上一栅格点只能位于点 $p$ 正下方.

A*算法中的栅格点状态分为new(初始化状态)、open(待扩展状态)和close(已扩展状态). 新算法中引入了一种新的状态incon[19], 称为不一致状态. 因此新算法在逻辑上包含3个队列: Open_list, Close_list和Incon_list. 本文为每个栅格点 $p$ 均绑定一个状态标识, 通过该状态标识, 可以直接识别栅格所在的队列, 从而免于队列查询. 本文中, Open_list采用一个顶端最小的有序二叉树, 排序依据为各个栅格的fval值, Incon_list采用一个后进先出的无序栈. Close_list由于只涉及栅格查询操作, 因此不再需要.

节点选取规则: 每次循环中, 算法总是优先选取Incon_list中搜索节点(栅格)为当前最优待扩展栅格, 记为CBN; 只有当Incon_list为空时, 才选择Open_list中栅格为CBN. 选取Incon_list中栅格时无顺序要求, 而Open_list则必须取其中具有最小fval的栅格. CBN从原本所在队列移除, 并标记为已扩展状

态(close).

定向扩展规则:一般地,方向$i$左右两侧第$k$个方向编号分别为

$$\begin{cases}\text{left}(i,k)=\text{mod}(i-k+8,8),\\ \text{right}(i,k)=\text{mod}(i+k,8),\end{cases}\tag{4}$$

其中$\text{mod}(x,8)$表示整数$x$对8取余.

设$pi$为CBN在$i$方向上的紧邻栅格,根据转弯约束,栅格CBN处末端方向与方向$i$相差不超过$T$的路径才能延伸至$pi$,此过程称为$i$方向上的定向扩展. 因此,集合$\{\text{left}(i,T),\text{left}(i,T-1),\cdots,\text{right}(i,T-1),\text{right}(i,T)\}$称为方向$i$的有效父代方向集合.

最优更新规则:CBN尝试更新相邻栅格点时,应当以CBN在该栅格点的有效父代方向集合上的最优路径为基础进行更新. 对于CBN在$i$方向上的紧邻栅格$pi$,$\text{ming}=\min\{\text{CBN.gval}[j]\}$,其中$j\in\{\text{left}(i,T),\text{left}(i,T-1),\cdots,\text{right}(i,T-1),\text{right}(i,T)\}$.

如果$\text{ming}+\text{octile}(\text{CBN},pi)<pi.\text{gval}[i]$,则称CBN在定向扩展规则下能够更新栅格$pi$,更新后栅格$pi$处第$i$方向路径为$pi.\text{gval}[i]=\text{ming}+\text{octile}(\text{CBN},pi)$. 如果此时$pi.\text{gval}[i]+pi.\text{hval}<pi.\text{fval}$,则总启发式代价$pi.\text{fval}$也被更新.

如果$pi$是未赋值的初始化点(new),或者$pi$是待扩展点(open)且能够被CBN更新,则更新栅格$pi$,且$pi$移入Open_list.

如果$pi$是已扩展点(close)或不一致点(incon),且能够被CBN更新,则更新栅格$pi$,并将$pi$移入Incon_list.

### 3.2 算法流程

假设当前栅格尺度下机器人转弯能力等级为$T$,任务要求的路径起始角方向编号为SA,目标角为GA,$\text{SA},\text{GA}\in\{0,1,\cdots,7\}$,则基于方向约束的A*算法流程如下:

Step 1: 初始化. 清空待扩展队列Open_list和不一致队列Incon_list,所有栅格点初始状态标识为new.

Step 2: 处理起始点$S$. 起始点$S$向SA方向进行定向扩展,更新其在SA方向上的相邻栅格,并将其放入Open_list. 起始点标记为close状态.

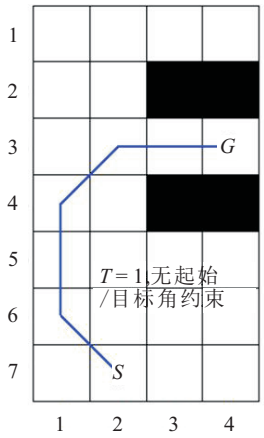
Step 3: 选取扩展点. 根据节点选取规则选取当前最优待扩展栅格CBN. 如果Incon_list和Open_list均为空,则路径规划失败,不存在满足要求的可行路径. 如果$G.\text{gval}[\text{GA}]\leqslant\text{CBN.fval}$,则最优路径已找到,成功结束,$G.\text{gval}[\text{GA}]$为结果路径长度.

Step 4: 扩展和更新. CBN对其每个相邻栅格(障碍栅格除外)进行定向扩展,并根据最优更新规则,尝试更新相邻栅格. 此过程中,CBN变为已扩展状态(close),新栅格进入Open_list或Incon_list,然后返回Step 3,直到算法结束.

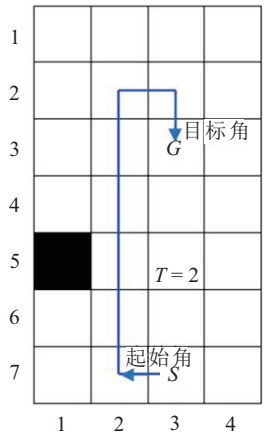
### 3.3 结果路径

算法结束后,最优路径通过回溯获得.

目标点$G$处的路径方向为GA,因此$G$的父节点位于$G$的$\text{mod}(\text{GA}+4,8)$方向,假设该栅格节点为$p$. 从$p$向$G$的运动方向为GA,由于算法遵循定向扩展机制,所代表的最优路径必然是$p$处方向集$\{\text{left}(\text{GA},T),\text{left}(\text{GA},T-1),\cdots,\text{right}(\text{GA},T-1),\text{right}(\text{GA},T)\}$各个路径中最短的那一条,根据该路径方向,即可确定栅格$p$的父节点. 重复以上步骤,直到回溯至起始点$S$. 图5给出了新算法下图3和图4中算例的结果路径.




(a) 图3算例




(b) 图4算例

**图5 基于方向约束的A*算法结果路径**

另外,如果任务起始角是若干方向组成的集合,即$\text{SA}\subset\{0,1,\cdots,7\}$,则只需在算法流程(小节3.2)的Step 2中,起始点$S$向SA所包含的所有方向上扩展子节点. 如果目标角$\text{GA}\subset\{0,1,\cdots,7\}$,则终止条件改为$\min\{G.\text{gval}[i]|i\in\text{GA}\}\leqslant\text{CBN.fval}$. 回溯结果路径时,取GA方向集内最短的那一条路径.

## 4　最优性证明

设$\pi = \{p0, p1, \cdots, pn\}$为从$S$到$G$,满足机器人转弯能力约束和任务起始角/目标角要求的一条栅格最优路径,路径点紧邻且顺序排列,$p0 = S, pn = G$. $A = \{a_1, a_2, \cdots, a_n\}$为机器人沿$\pi$运动时,抵达各个路径点时的速度方向(起始点$S$无父节点和抵达方向). 搜索过程中,对于最优路径上任一点$pm$,如果$pm.\text{gval}[am]$等于沿路径$\pi$从$S$到$pm$的长度,则称$pm$是最优点,显然此时

$$pm.\text{fval} \leqslant pm.\text{gval}[am] + pm.\text{hval} \leqslant \pi.\text{length}, \quad (5)$$

其中$\pi$.length为路径$\pi$的长度.

**定理1**　任意时刻,$\pi$上总是至少有一个最优点处于Open_list或Incon_list中.

**证明**　$p1$紧邻$S$,初始扩展$S$时,必然有

$$p1.\text{gval}[a_1] = \text{octile}(p1, S),$$

显然是沿路径$\pi$从$S$到$p1$的最优状态,此时$p1$被放入Open_list中.

当被$p1$扩展时

$$p2.\text{gval}[a_2] = p1.\text{gval}[a_1] + \text{octile}(p1, p2),$$

由于$p1.\text{gval}[a_1]$是沿$\pi$从$S$到$p1$路径长度,显然$p1.\text{gval}[a_1] + \text{octile}(p1, p2)$是沿$\pi$从$S$到$p2$路径长度,故$p2$是最优点. 如果$p2$此前为close或incon状态,则其将会被放入Incon_list,否则将会被放入Open_list.

以此类推,当最优点$pm$被扩展时,$pm+1$必然被更新为最优状态,并放入Open_list或Incon_list中. □

**定理2**　基于方向约束的A*算法总是能找到满足方向约束的最优栅格路径,除非它不存在.

**证明**　只要最优路径$\pi$存在,定理1就能保证Open_list$\bigcup$Incon_list总是不为空. 假设算法找到的结果路径为$\pi_0$,算法退出时的当前最优的待扩展栅格为CBN,则定向扩展机制下$\pi_0$必然满足方向约束,现讨论其最优性. 假设$\pi_0$不是最短路径,即

$$\pi_0.\text{length} > \pi.\text{length}. \quad (6)$$

情况1: 如果CBN属于Open_list,则根据算法退出条件,$\pi_0.\text{length} = G.\text{gval}[\text{GA}] \leqslant \text{CBN.fval}$. 显然,此时Incon_list为空,根据定理1, Open_list中必然存在一个最优点$pm$,且$pm.\text{fval} \leqslant \pi.\text{length} < \pi_0.\text{length} \leqslant \text{CBN.fval}$,即存在$pm$的fval值比CBN小,这与CBN的定义矛盾,因此此情况不能成立.

情况2: 如果CBN属于Incon_list. 假设最后一个被扩展的Open_list点为$ps$,则显然$ps$被扩展时,Incon_list为空,且根据定理1,此时Open_list中应当含有一个最优点,而此时$ps$具有Open_list中最小的fval,结合式(5),显然有

$$ps.\text{fval} \leqslant \pi.\text{length}. \quad (7)$$

另外,此时还应当有

$$ps.\text{fval} < G.\text{gval}[\text{GA}]. \quad (8)$$

否则算法立刻结束,回到情况1.

由于早于$ps$被扩展,此时任一close状态栅格$pc$均满足

$$pc.\text{fval} \leqslant ps.\text{fval}. \quad (9)$$

此后,所有的CBN均取自于Incon_list,不再有Open_list点变为close状态,因此Incon_list中所有的栅格(包括CBN)均取自于最初的close集合,而算法过程中,任何栅格点的fval都不会增大,结合式(9),满足算法终止条件时,必然有

$$G.\text{gval}[\text{GA}] \leqslant \text{CBN.fval} \leqslant ps.\text{fval}. \quad (10)$$

由式(8)和(10)可知,$ps$被扩展后,Incon_list元素被扩展的过程中,$G$被某incon栅格更新,$G.\text{gval}[\text{GA}]$值变小.

考虑$ps$将其紧邻close栅格点(记为$p'$)纳入Incon_list的过程,显然$ps$中某个方向(记为$k$)的路径更新优化了$p'$中的一条路径(记为$i$). 更新后,$p'.\text{gval}[i] + p'.\text{hval} = ps.\text{gval}[k] + \text{octile}(ps, p') + p'.\text{hval} \geqslant ps.\text{gval}[k] + ps.\text{hval}$,即路径$ps.\text{gval}[k]$延伸至$p'.\text{gval}[i]$过程中,路径总估计代价不可能减少.

因为$p'$原本是close点,曾经被扩展过至少一次,而$p'.\text{gval}[i]$是$p'$中唯一的新路径信息,因此$p'$再次被扩展时,如果能更新优化其某一紧邻栅格点(记为$p''$),则$p''$中被更新的路径必然是$p'.\text{gval}[i]$的延伸. 同上,$p''$中该路径的总估计代价不低于$ps.\text{gval}[k] + ps.\text{hval}$,依次类推. 因此,既然在Incon_list元素被扩展的过程中,$G.\text{gval}[\text{GA}]$路径信息被更新,必然有$G.\text{gval}[\text{GA}] = G.\text{gval}[\text{GA}] + \text{octile}(G, G) \geqslant ps.\text{gval}[k] + ps.\text{hval} \geqslant ps.\text{fval}$. 结合式(10),显然算法终止时$G.\text{fval}[\text{GA}]$等于$ps.\text{fval}$,亦即$\pi_0.\text{length}$等于$ps.\text{fval}$. 结合式(7),有$\pi_0.\text{length} \leqslant \pi.\text{length}$,但这显然与式(6)的假设相矛盾,因此情况2不成立.

综上,$\pi_0$不是最短路径的假设不成立,故基于方向约束的A*算法总是能找到满足方向约束的最短栅格路径. □

## 5　算法复杂度

A*算法的运算代价可以表示为[20]: 节点扩展次数×(链表操作开销+更新相邻节点开销). 当问题规模增大时,需要扩展的节点相应增多,算法需要进行

更多次的扩展搜索,与此同时,由于Open_list长度变长,每次搜索的链表操作开销(包括链表查询,元素出/入和链表排序等)也相应增大. 因此,问题规模对A*算法复杂度的影响呈超线性增长. 新算法引入了姿态维度,每一个栅格点理论上对应8个代表不同方向路径的搜索节点,从而增大了解空间和问题规模. 为此,新算法采用节点合并和集体扩展策略,将同一栅格点处的8个不同方向路径合并成为同一个搜索节点,通过编号区分不同方向路径,从而使栅格点与搜索节点仍然一一对应,搜索节点数目与A*算法处于同一水平.

以octile距离计算栅格节点估计代价hval符合A*算法的一致性原则[21],标准A*算法中的每个栅格节点最多被扩展一次. 但新算法中,每个栅格只选取各个方向路径中最短长度作为Open_list的排序依据,一定程度上破坏了一致性原则,因此新算法解决同一问题所需要的扩展次数要比A*算法多. 为此,新算法采用了不一致队列Incon_list,当某个close栅格点能够被更新优化时,即表明发生了局部不一致,该栅格点将被移入Incon_list,而不是Open_list. 而作为一个后进先出的无序栈,Incon_list不需要查询、排序和位置移动操作,几乎没有运算开销,因此其效率远高于Open_list. 第4节已证明,只要优先扩展Incon_list栅格,算法总能保持最优.

当尝试更新相邻栅格节点时,A*算法每个栅格只有一条路径,而新算法需要选取$(2T+1)$个方向上的最短路径进行定向扩展,因此更新相邻节点的开销变大,而这是保持解空间完整性和算法最优性的必然代价.

## 6 实验结果

为验证新算法的性能,本文进行了仿真验证,测试数据取自于丹佛大学举办的Grid based path planning competition(GPPC)比赛[15,22]. GPPC始于2012年,为基于栅格的路径规划算法提供标准的测试地图集和测试算例集(同一地图上一对起始点和目标点称为一个算例),规定了算法测试接口和性能指标测量方法,并提供了各个算例的参考最优解(无方向约束情况下). GPPC测试集运行结果可以作为不同的路径规划算法间性能对比的依据,已得到了广泛的采用和认可[17-20].

GPPC所有的测试地图集和对应算例均可通过其官网http://movingai.com获取. 本文实验中选取了其中的Baldur's GateII (BG2)数据集,该数据集来源于同名电脑游戏,测试地图为该游戏场景的栅格化结果,共包含75张地图和93 160个算例,所有地图均为512×512(单位为栅格),所有算例均随机产生并按照无方向约束时最优路径的长度分簇均匀分布. BG2地图集的特点是障碍分布不规则,并含有许多45°倾斜轮廓,地图形状接近现实场景. 图6展示了其中的一个示例地图,黑色部分为障碍区域.

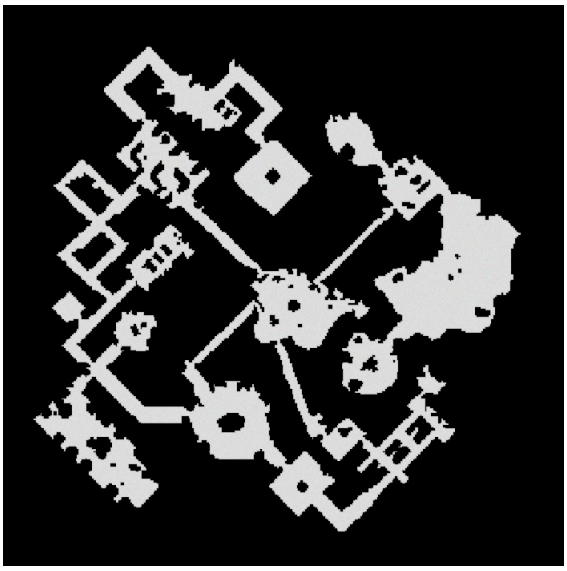


**图6 示例地图**BG2_AR0602SR

除了本文提出的基于方向约束的A*算法,本文还选取了对比算法. 针对机器人的转弯约束,实验对比算法选取了无约束的标准A*算法和禁止机器人非法转向的禁忌A*算法;针对起始/目标角约束,每种算法又分别采用临时起始/目标点和虚拟障碍区两种策略,因此共有4种对比算法. 实验中,分别选取机器人转弯能力等级$T=1$和$T=2$进行了两组实验,所有实验均在CPU型号Core i3,主频3.2 GHz,内存2 GB的PC机上进行,表1给出了实验结果.

需要注意的是,GPPC数据集原本是为无约束的栅格路径规划问题而设计. 考虑机器人方向约束时,其地图集和算例集仍然可以使用,但需要为每个测试算例额外添加起始/目标角约束,该约束随机产生并在不同算法间保持一致. 容易理解,添加方向约束后,部分原始算例不再拥有可行解. 如果某一算例能够被至少一种实验算法求解且结果路径满足方向约束,则称该算例为有效算例. 显然,转弯约束越苛刻,有效算例越少.

1) 完备性. 表1中列出了实验中,各个算法能够求解(即找到符合方向约束的结果路径)的算例数目,$T=1$时共有84 242个有效算例,$T=2$时共有86 843个有效算例. 可以看出,对所有有效算例,新算法均能成功求解. 不存在能被某一现有对比算法求解,但不能被新算法求解的算例. 相比之下,各

**表 1**　BG2测试数据集上的运行结果对比

| 转弯能力 | 算法 | 成功解决算例 | | 平均路径长度 | | 平均求解时间/ms | |
|---|---|---|---|---|---|---|---|
| | | 算例数目 | 成功率/% | 对比算法 | 新算法 | 对比算法 | 新算法 |
| $T=1$ | 基于方向约束的A* | 84 242 | 100 | — | 252.044 | — | 17.511 |
| | A*+临时起始/目标* | 9 503 | 11.3 | 239.695 | 239.695 | 3.399 | 13.478 |
| | A*+虚拟障碍* | 1 597 | 1.9 | 239.575 | 239.043 | 3.624 | 12.310 |
| | 禁忌A*+临时起始/目标 | 29 887 | 35.5 | 246.275 | 246.176 | 3.540 | 14.393 |
| | 禁忌A*+虚拟障碍 | 5 467 | 6.5 | 249.565 | 248.129 | 3.808 | 13.639 |
| $T=2$ | 基于方向约束的A* | 86 843 | 100 | — | 251.239 | — | 16.475 |
| | A*+临时起始/目标* | 31 245 | 36.0 | 247.118 | 247.118 | 3.812 | 15.202 |
| | A*+虚拟障碍* | 21 815 | 25.1 | 251.338 | 248.783 | 5.280 | 16.100 |
| | 禁忌A*+临时起始/目标 | 51 235 | 59.0 | 245.483 | 245.482 | 3.748 | 14.953 |
| | 禁忌A*+虚拟障碍 | 21 815 | 25.1 | 251.338 | 248.783 | 5.272 | 16.101 |

个对比算法只能求解一部分有效算例,均不具备全局求解能力,$T=1$时对比算法的求解成功率为1.9%$\sim$35.5%,$T=2$时为25.1%$\sim$59.0%. 显然,新算法求解方向约束路径规划问题的能力远高于现有算法,并能够始终保持解空间的完备性.

2) 最优性. 现有对比算法均只能求解有效算例的一部分,且各个对比算法能够求解的算例集合各不相同. 为验证新算法路径最优性,表1给出了各个算法在各自能够成功求解的有效算例集合上的平均结果路径长度以及平均求解时间.

容易理解,采用临时起始/目标点策略的标准A*算法并不损失解空间,只是未考虑方向约束,虽然求解成功率较低,但只要能够成功求解(即结果路径恰好满足方向约束),其给出的总是该算例下的最短路径.

表1结果表明,在相同有效算例集合上,新算法结果路径平均长度总是与采用临时起始/目标点策略的标准A*算法相同. 例如,$T=1$时,在采用临时起始/目标点策略的标准A*算法能够求解的9 503个算例上,两种算法结果路径的平均长度均为239.695. 显然,新算法能够保证给出最短长度路径. 而新算法能够求解所有的84 242个有效算例,这是现有算法做不到的. 其他3种对比算法都不具备最优性,表1表明,它们在相同算例集合上的结果路径都比新算法长.

3) 算法效率. 表1还表明,虽然引入了姿态维度,增大了解空间,但是通过采用节点合并策略和不一致队列Incon_list,新算法平均求解时间最多不超过A*算法的4倍,考虑到新算法具有现有算法所不具有的完备性和最优性,以及由此带来的求解成功率和路径质量上的提高,新算法能够满足工程实践需求.

## 7　结　论

为解决机器人路径规划问题中的转弯约束和起始/目标角约束,本文提出了一种基于方向约束的A*算法. 理论分析表明,新算法能保证给出符合转弯约束和起始/目标角约束的最短路径. 在典型地图集上与4种对比算法的比较结果表明,相比于现有算法,新算法能够有效提高方向约束路径规划问题的求解能力和路径质量.

本文采用了八向栅格图,容易证明,当方向划分更精细时(例如采用十六向栅格图),文中算法依然适用,但算法开销增大,因此下一步工作包括设计更高效的数据结构和定向扩展策略,以进一步提高算法效率.

## 参考文献(References)


[1] Sanchez A, Cuautle R, Zapata R, et al. A reactive lazy PRM approach for nonholonomic motion planning[J]. Lecture Notes in Computer Science, 2006, 4140: 542-551.

[2] 李红. 自动泊车系统路径规划与跟踪控制研究[D]. 长沙: 湖南大学机械与运载工程学院, 2014.
(Li H. A study on path planning and tracking control method for automatic parking system[D]. Changsha: College of Mechanical and Vehicle Engineering, Hunan University, 2014.)

[3] Liu J, He Y, Zhang A. An improved MULTI-ACS algorithm for the waste collection vehicle arc routing problem with turn constraints[C]. Proc of the 10th Int Conf on Computational Intelligence and Security. Kunming: IEEE Press, 2014: 25-29.

[4] Boroujerdi A, Uhlmann J. An efficient algorithm for computing least cost paths with turn constraints[J].

Information Processing Letters, 1998, 67(6): 317-321.

[5] 郭路生. 足球机器人路径规划研究与实现[D]. 武汉: 中国地质大学信息工程学院, 2006.
(Guo L S. Research and realization on path planning for soccer robots[D]. Wuhan: Faculty of Information Engineering, China University of Geosciences, 2006.)

[6] 王怿, 祝小平, 周洲. 一种基于Clothoid曲线的无人机路径规划算法[J]. 西北工业大学学报, 2012, 30(6): 874-878.
(Wang Z, Zhu X P, Zhou Z. An unmanned aerial vehicle path planning algorithm based on the clothoid curve[J]. J of Northwestern Polytechnical University, 2012, 30(6): 874-878.)

[7] 王荔豪, 王爱华, 柯楠, 等. 基于Dubins曲线的高超声速飞行器路径规划[J]. 弹箭与制导学报, 2015, 35(3): 41-43.
(Wang L H, Wang A H, Ke N, et al. Route planning for hypersonic vehicle based on dubins curves[J]. J of Projectiles, Rockets, Missiles and Guidance, 2015, 35(3): 41-43.)

[8] Tao C, Wang S, Tian G, et al. Integrated guidance and control with terminal impact angular constraint for bank to turn flight vehicle[C]. Chinese Control Conf. Nanjing: IEEE Press, 2014: 681-685.

[9] Wen Q, Xia Q, Su W. Bank-to-turn guidance law with terminal impact-angle constraint[J]. J of Aerospace Engineering, 2016, 29(3): 1-8.

[10] Zhang Q, Wang Z, Tao F. Optimal guidance law design for impact with terminal angle of attack constraint[J]. Optik-Int J for Light and Electron Optics, 2014, 125(1): 243–251.

[11] Tao C, Wang S, Yang M. Guidance with terminal impact angular constraint for hybrid BTT/STT flight vehicle[C]. Chinese Control Conf. Nanjing: IEEE Press, 2014: 583-588.

[12] 叶炜垚, 王春香, 杨明, 等. 基于虚拟障碍物的移动机器人路径规划方法[J]. 机器人, 2011, 33(3): 273-278.
(Ye W Y, Wang C X, Yang M, et al. Virtual Obstacles Based Path Planning for Mobile Robots[J]. Robot, 2011, 33(3): 273-278.)

[13] 雷兆明, 孙鹤旭, 刘作军, 等. 非完整性约束下移动机器人带虚拟障碍物的路径规划[C]. 中国控制会议. 北京: 北京航空航天大学出版社, 2007: 207-211.
(Lei Z M, Sun H X, Liu Z J, et al. Virtual obstacle based approach to path planning of mobile robot with nonholonomic constraints[C]. Chinese Control Conf. Beijing: Beihang University Press, 2007: 207-211.)

[14] 夏洁, 高金源. 满足战场需求的实时飞行路径规划[J]. 北京航空航天大学学报, 2004, 30(2): 95-99.
(Xia J, Gao J Y. Real-time flight path planning for combat mission[J]. J of Beijing University of Aeronautics and Astronautics, 2004, 30(2): 95-99.)

[15] Sturtevant N R. The grid-based path planning competition[J]. AI Magazine, 2014, 35(3): 66-69.

[16] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Trans on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[17] Botea A. Fast, optimal pathfinding with compressed path databases[C]. Proc of the 5th Annual Symposium on Combinatorial Search. Ontario: AAAI Press, 2012: 204-205.

[18] Botea A. Ultra-fast optimal pathfinding without runtime search[C]. Proc of the Seventh AAAI Conf on Artificial Intelligence and Interactive Digital Entertainment. Menlo Park: AAAI Press, 2011: 122-127.

[19] Likhachev M, Gordon G J, Thrun S. ARA*: Anytime A* with provable bounds on sub-optimality[C]. Proc of 2003 Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2003: 767-774.

[20] Harabor D, Grastien A. Improving jump point search[C]. Proc of the 24th International Conf on Automated Planning and Scheduling. Menlo Park : AAAI Press, 2014: 128-135.

[21] Daniel K, Nash A, Koenig S, et al. Theta*: any-angle path planning on grids[J]. J of Artificial Intelligence Research, 2010, 39(1): 533-579.

[22] Sturtevant N R. Benchmarks for grid-based pathfinding[J]. IEEE Trans on Computational Intelligence and AI in Games, 2012, 4(2): 144-148.

（责任编辑：齐 霁）