基于搜索的算法调研综述

1. **研究背景**

在无人驾驶技术的研究中，搜索算法被广泛应用于路径规划和避障等问题中。搜索算法是一种通过在问题状态空间中搜索解决方案的算法，它能够快速有效地找到最优解。搜索算法在无人驾驶技术中的应用能够提高车辆行驶的安全性和效率，降低驾驶员的负担和疲劳程度，进一步推动无人驾驶技术的发展和应用。

因此，无人驾驶基于搜索算法的研究具有重要的背景和意义，能够为实现智能交通系统和无人驾驶技术的普及和推广做出贡献。

1. **研究现状**（对几篇论文进行讲解）

路径搜索算法是一种基于地图，在起点和目标点之间搜索路径的方法。而在路径搜索算法中a -star算法是在Dijkstra算法是使用最多的算法，由于他们提出的时间比较早，在现如今的研究中都存在一些缺陷，不能适应现如今的研究要求。因此进年的研究都对这两种算法进行了改良

1. **算法基本思路**
2. Dijkstra算法
3. 初始化：

将起始节点标记为当前节点，并设置起始节点的最短路径为0。  
 将其他节点的最短路径初始化为无穷大（表示尚未计算出最短路径）。  
 创建一个空的集合用于存储已经确定最短路径的节点。

1. 重复以下步骤直到找到目标节点或所有可达节点的最短路径都被确定：

从未确定最短路径的节点中选择一个最短路径最小的节点作为当前节点。  
 将当前节点标记为已确定最短路径，并将其加入到确定最短路径的节点集合中。

1. 对于当前节点的所有邻居节点，更新其最短路径：

计算通过当前节点到达邻居节点的路径长度，将其与邻居节点的当前最短路径进行比较。  
 如果通过当前节点到达邻居节点的路径长度更短，更新邻居节点的最短路径为新的路径长度

1. 重复步骤2和步骤3，直到所有可达节点的最短路径都被确定。
2. 如果目标节点被确定最短路径，算法结束。否则，表示目标节点无法从起始节点到达，无最短路径。
3. 如果需要找到最短路径，可以从目标节点逆向追溯，通过节点的前驱指针或其他方式，得到从起始节点到目标节点的最短路径。

（2）A\*算法

1、定义两个集合：Open集合和Closed集合。Open集合存储待扩展的节点，Closed集合存储已经扩展过的节点。

2、将起始节点加入Open集合，并设置其初始代价（例如起始节点到目标节点的估计距离）。

3、重复以下步骤直到找到目标节点或Open集合为空：  
 从Open集合中选择代价最小的节点作为当前节点。  
 如果当前节点是目标节点，搜索结束，找到了最短路径或最佳路径。  
 否则，将当前节点从Open集合移入Closed集合，并扩展当前节点的邻居节点。  
4、对于每个邻居节点，计算其代价函数值（一般是当前节点到邻居节点的实际代价加上邻居节点到目标节点的估计代价）。

如果邻居节点已经在Closed集合中，跳过该节点。  
 如果邻居节点不在Open集合中，将其加入Open集合，并更新其代价函数值。  
 如果邻居节点已经在Open集合中，并且新的代价函数值比之前的小，更新邻居节点的代价函数值。

5、如果Open集合为空，表示无法找到路径，搜索失败。

6、如果搜索成功，可以从目标节点逆向追溯，得到最短路径或最佳路径。

**2、算法的改进**

（1）改进的Dijkstra算法

在Guo Qing[1]的文章中提出了一种改进的Dijkstra算法，它在路径搜索时保存所有的等距最短路径，根据转弯时间从若干条等距路径中选择最优路径，从而得到距离和时间都最短的路径。

由于现在的规划都是用网格图，因此相同距离的最短路径可能不止一条。而传统的Dijkstra算法只能找到一条的最短路径对Dijkstra算法进行了改进，使其能够在矩形环境映射中找到所有最短路径。

这篇文章对Dijkstra算法进行了改进，使其能够在矩形环境映射中找到所有最短路径。Dijkstra算法只存储一个中间节点，通过搜索只能找到一条最短路径。文中改进了Dijkstra算法，保留所有与源节点距离相同的节点作为中间节点，并从所有中间节点重新搜索，直到遍历到目标节点。通过多次迭代，找到所有距离相同的最短路径。

文章需要解决最短的行驶时间，而AGV(Automated Guided Vehicle)在距离相同的路径情况下转弯个数少的路径即为要找到的最短的距离和最短的形式路径，文中对于环境图进行网格化并编号通过判断节点是否为转弯节点，然后利用改进的Dijkstra算法计算每条最短路径的转弯节点数，从而找到转弯最少的路径。

（2）改进的A\*算法

目前，改进A\*算法的方法主要有两种。第一个基于预处理步骤，而第二个应用后处理。预处理方法是在算法搜索过程中对路径进行优化。

一种改进的A\*算法，通过判断当前路径点与目标点之间是否存在障碍物来缩短路径。然而，这种方法计算量大，并且不能使转弯路径平滑。后处理方法是在算法搜索结束后对路径进行优化。

一种路径增强方法，通过判断相邻路径节点之间的直线是否经过障碍物来缩短路径长度。但是，这会产生更大的转弯角度。然后采用SPS算法生成障碍物周围的点集，然后通过粒子群算法不断更新节点的位置。

这篇文章[2]一些改进算法是基于A\*算法，如双向A\*算法，该算法同时搜索起始点和目标点。最后，宽度优先搜索(BFS)和深度优先搜索(DFS)算法在网格地图中也得到了广泛的应用。BFS算法是使用队列实现的搜索算法，DFS算法是使用递归实现的搜索算法。本文介绍了一种基于几何A\*算法的路径规划方法。为了避免传统A\*算法产生的锯齿形路径和交叉路径中存在的节点多、距离长、转弯角度大等问题，将该方法应用于自动导引车辆(AGV)。首先，采用网格方法对端口环境进行建模。其次，通过函数P (x, y)和W (x, y)对闭合列表中的节点进行过滤，去除不符合要求的节点，避免产生不规则路径。

1. **述评**

路径搜素算法根据策略的不同，路径搜索算法可分为仿生算法、几何模型搜索算法和基于插值的算法。

仿生算法是一种模拟自然生物进化或群体社会行为的随机搜索算法。粒子群算法(PSO)、遗传算法(GA)、蚁群算法(AC)等一系列算法已被应用于路径搜索过程。虽然它们通常很容易实现，但它们通常计算成本很高，并且经常陷入一些局部最优，因此它们并不总是适用于复杂网格地图环境中的路径规划。

基于插值的算法一般采用平滑曲线，如b样条曲线]和dubins曲线来拟合整个路径。生成的路径光滑且具有连续曲率，这是满足速度和加速度要求的宝贵属性。但它们并不适用于不规则的路径，也常常局限于局部平滑或简单的路径问题。

几何搜索算法是常用的路径搜索算法，主要包括Dijkstra算法和A\*算法。Dijkstra算法通过遍历所有节点获得最短路径。该方法获得路径的成功率高，但计算效率较低。A\*算法是在Dijkstra算法的基础上加入启发式函数的一种搜索算法。与Dijkstra算法相比，A\*算法在求解最短路径问题时更简单、更直接，但该算法也容易陷入局部最优。

这里主要讨论A\*算法是在Dijkstra算法。

1. Dijkstra算法

Dijkstra算法是解决最短路径问题的经典算法。它可以从一个节点得到到其他每个节点的最短路径。其时间复杂度为O(n2)。n是图中节点的个数。这个最小距离不仅可以用纯长度来衡量，还可以用其他方式来衡量，例如时间开销、经济成本、吞吐量等。

Dijkstra算法简单易行，是路径规划的一种优秀方法，已广泛应用于网络优化、交通运输、物流、电子等领域。(GPS导航系统一般采用这种方法进行路径规划)。路径规划包括全局路径规划和局部路径规划，Dijkstra算法属于前者。所以它可以保证生成的路径是最短路径。

Dijkstra算法的特点是最短路径的任何子路径都是其起始节点到终端节点的最短路径，因此它可以获得路径中从源节点到目标节点的最短路径。

而它的缺点在于；

1. Dijkstra算法只考虑了行程距离而不考虑行程时间，因此最短路径可能不满足最短行程时间的要求。这种解决方案仍然存在时空开销与精度的矛盾。
2. Dijkstra算法是基于无向图的。得到的路径依赖于无向图的拓扑结构。因此，算法的效果取决于无向图结构。当地图发生改变时需要重新设计无向图，再对路径进行规划，工作繁琐。

2、A\*算法

A\*算法是一种应用广泛的启发式搜索算法。A\*算法采用深度优先遍历规则和启发式算法，算法收敛速度快。不仅在路径搜索过程中运行速度快，而且具有良好的实时性，因此在路径规划中得到了广泛的应用。

而它的缺点在于；

当A\*算法用于复杂环境下的路径规划时，会产生交叉路径和锯齿路径。如果地图尺寸太大或环境太复杂，A\*算法将严重影响的效率。

1. **文献参考**

[[1]Qing G, Zheng Z, Yue X. Path-planning of automated guided vehicle based on improved Dijkstra algorithm[C]//2017 29th Chinese control and decision conference (CCDC). IEEE, 2017: 7138-7143.](http://www.jkly.xyz:70/doc-delivery/file/download/5348100)

[[2]Tang G, Tang C, Claramunt C, et al. Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment[J]. IEEE access, 2021, 9: 59196-59210.](http://spis.hnlat.com/scholar/downloadLog?link=https%3A%2F%2Fieeexplore.ieee.org%2Fiel7%2F6287639%2F9312710%2F09391698.pdf&title=Geometric%20A-star%20algorithm%3A%20An%20improved%20A-star%20algorithm%20for%20AGV%20path%20planning%20in%20a%20port%20environm)

[[3]Zhang Z, Zhao Z. A multiple mobile robots path planning algorithm based on A-star and Dijkstra algorithm[J]. International Journal of Smart Home, 2014, 8(3): 75-86.](http://spis.hnlat.com/scholar/downloadLog?link=https%3A%2F%2Fgvpress.com%2Fjournals%2FIJSH%2Fvol8_no3%2F7.pdf&title=A%20multiple%20mobile%20robots%20path%20planning%20algorithm%20based%20on%20A-star%20and%20Dijkstra%20algorithm&_=0.7654337868118264)

[[4]Wang C, Wang L, Qin J, et al. Path planning of automated guided vehicles based on improved A-Star algorithm[C]//2015 IEEE International Conference on Information and Automation. IEEE, 2015: 2071-2076.](http://www.jkly.xyz:70/doc-delivery/file/download/5348091)

[[5]Zheng T, Xu Y, Zheng D. AGV path planning based on improved A-star algorithm[C]//2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE, 2019: 1534-1538.](http://spis.hnlat.com/scholar/redirect?url=https%3A%2F%2Fieeexplore.ieee.org%2Fabstract%2Fdocument%2F8983841%2F&title=AGV%20path%20planning%20based%20on%20improved%20A-star%20algorithm)

1. [XiangRong T, Yukun Z, XinXin J. Improved A-star algorithm for robot path planning in static environment[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1792(1): 012067.](http://spis.hnlat.com/scholar/downloadLog?link=https%3A%2F%2Fiopscience.iop.org%2Farticle%2F10.1088%2F1742-6596%2F1792%2F1%2F012067%2Fpdf&title=Improved%20A-star%20algorithm%20for%20robot%20path%20planning%20in%20static%20environment&_=0.5397892442823196)

[7][Wang H, Yu Y, Yuan Q. Application of Dijkstra algorithm in robot path-planning[C]//2011 second international conference on mechanic automation and control engineering. IEEE, 2011: 1067-1069.](http://spis.hnlat.com/scholar/redirect?url=https%3A%2F%2Fieeexplore.ieee.org%2Fabstract%2Fdocument%2F5987118%2F&title=Application%20of%20Dijkstra%20algorithm%20in%20robot%20path-planning)