

文章编号: 1002-0446(2005)03-0210-05

多 AGV 调度系统中的两阶段动态路径规划*

刘国栋^{1, 2, 3}, 曲道奎^{1, 2}, 张雷^{1, 2}

(1. 中国科学院沈阳自动研究所, 辽宁 沈阳 110016; 2. 沈阳新松机器人自动化股份有限公司, 辽宁 沈阳 110016;
3. 中国科学院研究生院, 北京 100039)

摘要: 为缩短 AGV 系统的研发周期、降低研发成本, 寻求能适应不同地图的通用调度策略以及增强对实际调度任务和故障的鲁棒性, 提出了一种两阶段动态路径规划策略。对多 AGV 调度系统应用两阶段控制策略: 采用动态路径规划进行路径生成, 实时对多个 AGV 同时规划其路径, 并通过启发式算法实现路径优化。通过系统仿真证明, 该策略很好地提高了 AGV 调度系统的柔性, 是一种能适用于不同地图的通用调度策略。

关键词: 多 AGV 调度系统; 两阶段控制策略; 动态路径规划; k 最短路径; 启发式算法

中图分类号: TP24 **文献标识码:** B

Two-stage Dynamic Path Planning for Multiple AGV Scheduling Systems

LIU Guo-dong^{1, 2, 3}, QU Dao-ku^{1, 2}, ZHANG Lei^{1, 2}

(1. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China;
2. SIASUN Robot & Automation Co., Ltd., Shenyang 110016, China;
3. Graduate School of the Chinese Academy of Sciences, Beijing 100039, China)

Abstract: In order to shorten the R&D period of AGV (automated guided vehicle) systems, reduce the cost of R&D, find out universal scheduling strategy for different maps, and improve robustness to actual scheduling tasks and malfunc, a two-stage dynamic path planning strategy is developed. Multiple AGV scheduling systems are controlled with the two-stage control strategy, and paths are acquired by dynamic path planning. Several AGVs' paths are planned simultaneously in real time and path optimization is achieved with a heuristic algorithm in dynamic path planning. It is proved with simulation that the strategy improves flexibility of AGV systems and is a universal scheduling strategy for different maps.

Keywords: multiple AGV scheduling system; two-stage control strategy; dynamic path planning; k -shortest path; heuristic algorithm

1 引言 (Introduction)

随着物流技术的广泛应用, 自动导引车 (AGV) 在车间物流系统中成为实现敏捷物料处理的重要工具。因此, 多 AGV 调度系统, 也越来越受到广大学者的关注^[1-3]。但大部分学者是将多 AGV 调度系统作为多智能体或离散事件动态系统, 进行偏向理论的系统研究和建模, 而在实际的 AGV 研发中, 缩短研发周期、降低研发成本, 较为通用的调度策略及对实际调度任务和故障的鲁棒性显得尤为重要。本文用有向图^[4]对 AGV 运行轨迹建模, 提出了一种针对细化后的子任务的集中式调度策略^[4]——两阶段动态路径规划策略进行路径规划, 实现上层任务规划对路

径规划的透明。其中, 采用了两阶段控制策略^[5]对多 AGV 系统进行控制, 它的特点是受地图结构变化的影响小, 并且大大减小了实时在线运算的负担; 同时结合动态路径规划策略, 通过任务触发的模式, 实时对多个 AGV 进行路径规划。

2 多 AGV 调度系统 (Multiple AGV scheduling system)

图 1 描述了一个多 AGV 调度系统^[1]的地图配置简图。主要工作流程是将工件从仓库运送到几条并行的检测线上检测, 并将检测后的工件根据检测合格与否的结果运送到相应的站点。

AGV 的任务可描述为：位于起始节点 s (表示为 n_s) 的 AGV, 派发到目标节点 d (表示为 n_d)。这里不考虑任务的性质 (例如取货、送货或空闲调度等), 由上层的任务机制来优化和细化任务, 这样可以使在 AGV 的路径规划中只存在单一类型的任务 (最细化的子任务), 便于实现地图配置的通用性, 实现上层任务对下层路径规划的透明问题描述:

Fig. 1 Guide path model

本文采用双向图为 AGV 系统建模, 这里假定:

1) AGV 系统配置网络是一个允许具有强联接子网络的网络。强联接网络即在任何两个节点间存在至少两条不同的路径。

2) AGV 的数量少于节点数。

对一个 AGV 的路径规划就是寻找最短路径, 使 AGV 以最短的时间完成任务。当多 AGV 在地面上运行时, 对 AGV 的路径寻优, 实现 AGV 的总体运行时间最短; 并考虑阻塞程度, 用延迟等待时间对其描述, 使其等待时间最短; 还要求所得路径是无死锁的。

3 两阶段控制策略 (Two-stage control strategy)

两阶段控制策略, 又称两阶段交通控制方案^[5], 是先离线生成候选路径集, 再根据候选路径集进行总体路径规划的分阶段路径规划策略, 总体结构见图 2。

控制器的总体结构分为两个相继的模块, 离线的路径表生成器和在线的交通控制器。

首先, 根据导向路段的配置约束确定诱导网络模型。用该模型和改进的 k 最短路段算法, 离线路径表生成器 (RTG) 离线生成从每个站点到所有其它站

点的 k 条候选路径集, 并以路径表的形式存储起来, 构成路径库。当有任务下达时, 在线交通控制器 (OTC) 利用路径库提供的路径表, 再根据正在执行任务的 AGV 的运行信息, 生成无碰撞的优化路径。

Fig. 2 Overall structure of the two-stage control strategy

3.1 离线阶段——路径库的生成

(1) 抽取导向路段信息

即获取包括站点节点和辅助节点 (辅助节点是为了便于路径规划, 描述 AGV 运行轨迹所设的节点) 在内的各节点之间的无向图网络模型。图 1 所示模型即为抽取导向路段信息后所建立的模型的图示表示。

(2) 建立诱导网络模型

诱导网络模型是一种有向图模型, 它考虑了 AGV 运行的约束信息, 反映了 AGV 从任意节点出发到达其它节点的所有可能路径。

(3) k 最短路径搜索

对诱导网络模型进行搜索, 寻求从节点 s 到节点 d 的 k 条最短路径。便于路径库生成的 k 最短路径算法^[5,6]如下:

初始化: 设定 $k=0$, k 最大值 k_{max} 为 C 。

步骤 1: 设定起始点 s_0 , 采用经典的 Dijkstra 算法^[7], 求得 s 到其它任意节点的最短路径;

步骤 2: 历遍所有节点作为起始点 s_0 , 存储在路径库中;

步骤 3: 设定起始节点 s 及终止节点 d , 权值 $D = \infty$, 历遍所有节点作为 s 和 d , 跳转到步骤 7;

步骤 4: 从 $s \rightarrow d$ 的最短路径及次最短路径中选取中间节点 m (含 s 和 d);

步骤 5: 检索 m 的扩展节点到终止节点 d 的最短路径及次最短路径, 如果路径中含有重复节点, 或路径长大于 D , 则检索下一节点, 否则 $D = \text{路径长}$, 记录所得路径; 如果扩展节点的某一级最短路径的路径长大于 D , 则不需计算下一级最短路径。检索完毕后, 跳转到步骤 4;

步骤 6: $k++$, 记录所得路径为第 k 条最短路径. 如果 $k < k_{\max}$, 则 $D = \infty$, 跳转到步骤 4; 如果 $k \geq k_{\max}$, 则跳转到步骤 3;

步骤 7: 算法结束.

4) 路径表

路径表是路径库中数据元的存储形式, 它描述了从起始节点到终止节点的 k 条最短路径的节点序列. 在在线阶段也存在路径表, 它是生成优化轨迹的主要数据源, 但与路径库中的路径表略有不同, 具有从当前时刻发起的沿路径的时间轴, 称之为即时路径表.

3.2 在线阶段——优化路径的生成

在线阶段, OTC 包含两部分:

(1) 动态路径生成器;

(2) 监控器: 在对 AGV 实时控制时, 确保 AGV 按生成路径运行, 避免碰撞; 并对异常产生报告, 提交给控制台. 监控器的存在, 使对 AGV 的控制形成了闭环, 构成了反馈控制. 本文不对监控器做详细介绍.

4 动态路径规划 (Dynamic path planning)

动态路径规划是在新任务触发下, 先根据 AGV 当前运行信息产生报表, 报表描述正在承担任务的 AGV 从即时位置到目标节点的信息, 它与新任务一起作为任务需重新规划, 然后从路径库调入即时路径表, 对多个 AGV 同时规划其路径, 生成无冲突、总体运行时间最短的优化路径.

4.1 动态路径规划的复杂性分析

每次规划都是对 n 个 AGV 同时进行规划, 而每个 AGV 承担一个任务, 即相当于 n 个任务同时规划, 每个任务又有 k 条候补路径. 最坏情况下, 需对 n 个任务的所有 k 条路径的组合进行评估, 所以需要 n^k 次计算, 属于 NP 问题.

4.2 冲突检测

在多 AGV 调度系统中存在两种类型的冲突^[8]: 相向冲突和超前冲突. 对冲突的检测如下:

(1) 检测冲突的空间可能性: 检测 AGV 运行路径是否存在重叠路段;

(2) 检测冲突的时间可能性: 检测 AGV 在重叠路段上是否存在时间上的重叠点或重叠段;

(3) 判定冲突类型: 根据重叠路段上的运行方向, 判定冲突类型.

4.3 优化目标及其计算

对多个 AGV 整体进行优化, 即实现总体运行时

间最短. 总体运行时间定义为所有 AGV 在各自给定路径上运行所需时间的最大者. 见 4.4 节中的算法.

为了便于描述, 定义符号、变量如下:

设 i, m, n 为小车编号, $i, m, n \in I = \{i: i=1, 2, \dots, N\}$, N 为小车数; T_i 为第 i 辆小车承担的任务; $P(T_i)$ 为任务 T_i 所确定的可选路径集, $P(T_i) = \{j: j=1, 2, \dots, k < k_{\max}\}$; j, j_1, j_2 为路径编号, $j \in P(T_i)$; S 为调度方案, $S = (j, j_1, \dots, j_k)$; $R_i(j)$ 为第 i 辆小车按照其所承担任务的第 j 条路径运行的理想运行 (不存在等待延迟) 时间; $RP_i(S)$ 为在调度方案 S 下, 第 i 辆小车所行路径与其它小车所行路径存在的重叠路段的有序集合, $RP_i(S) = \{\eta_i^t: 0 \leq t \leq t_i, t \in Z^+\}$, t_i 为重叠路段数; $ST_i^t(S)$ 为在调度方案 S 下, 第 i 辆小车在第 t 段重叠路段上理想运行时的起始时间; $ET_i^t(S)$ 为在调度方案 S 下, 第 i 辆小车在第 t 段重叠路段上理想运行时的终止时间; $RST_i^t(S)$ 为在调度方案 S 下, 第 i 辆小车在第 t 段重叠路段上实际运行时的起始时间; $RET_i^t(S)$ 为在调度方案 S 下, 第 i 辆小车在第 t 段重叠路段上实际运行时的终止时间; $C(\eta_i^t)$ 为在路段 η_i^t 上与相互冲突的车辆编号按 $RST_i^t(S)$ 升序排列的有序集合; $O_i^t(S)$ 为在调度方案 S 下, 第 i 辆小车在第 t 段重叠路段上的等待延迟时间; $D_i(t)$ 为第 i 辆小车在经过第 t 段重叠路段后的累计等待延迟时间, 表达式为:

$$O_i^t(S) = D_i(t) - D_i(t-1) \quad (1)$$

$$RST_i^t(S) = ST_i^t(S) + D_i(t-1) \quad (2)$$

$$RET_i^t(S) = ET_i^t(S) + D_i(t) \quad (3)$$

小车的实际运行时间:

$$RR_i(j) = R_i(j) + \sum_{t=1}^{t_i} O_i^t(S) \quad (4)$$

总体运行时间:

$$TRT = TRT(S) = \max_{i \in N} (RR_i(j)) \quad (5)$$

等待时间:

$$WT = WT(S) = \sum_{i=1}^N \sum_{t=1}^{t_i} (O_i^t(S)) \quad (6)$$

总体优化目标函数为:

$$\min F(j, j_1, \dots, j_k) = \min (TRT) \quad (7)$$

阻塞函数^[9]反映了 AGV 在结构化地图上运行的阻塞程度, 其式为:

$$G = G(S) = \frac{WT}{\sum_{i=1}^N RR_i(j)} \quad (8)$$

优化目标的计算步骤如下:

- 1) 得到重叠路段集 $RP_i(S)$.
- 2) 沿时间轴方向,在起始时间 $RST_i^t(S)$ 下为重叠路段排序.
- 3) 依次对重叠路段进行冲突检测:
 - ⊗ 首先对经过重叠路段的 AGV,按到达时间先后排序 $C(\eta_i^j)$;

$$O_i^t(S) = \begin{cases} 0 & RST_i^t(S) < RST_j^t(S) \cup RST_i^t(S) - RST_j^t(S) > A \\ A - (RST_i^t(S) - RST_j^t(S)) & 0 < RST_i^t(S) - RST_j^t(S) < A \end{cases}$$

A是根据 AGV的车长给定的常数; j 为与之相冲突的 AGV编号; t 为相应的重叠路段序号.

如果为相向冲突,则:

$$O_i^t(S) = \begin{cases} 0 & RST_i^t(S) < RST_j^t(S) \\ RET_i^t(S) - RST_i^t(S) & RST_i^t(S) > RST_j^t(S) \end{cases}$$

- 4) 计算 $D_i(t)$ 、 $RST_i^t(S)$ 、 $RET_i^t(S)$,若扫描完所有重叠路段,则到 5);否则,回到 2).
- 5) 计算总体运行时间 TRT和阻塞函数 G.

4.4 求解算法

算法主要考虑如下问题:

- 1)问题的复杂度为 NP;
- 2)优化目标函数具有非线性;
- 3)求解的最优解应是无冲突的.

考虑到以上问题,本文提出了根据阻塞函数划定搜索域的启发式算法,该法降低了问题复杂度,加快了搜索最优解的速度.首先,将 k 条候补路径按照路径长度升序排列.这样最优解存在的概率随搜索深度的增加而递减.然后,采用由上至下、广度优先的搜索策略,计算各可行解的优化目标函数值.在搜索时,通过阻塞门限划定搜索深度.考虑到优化目标函数的非线性主要受阻塞程度影响,当阻塞函数为零时,即在 AGV运行时不存在等待延迟,再继续向下搜索将没有意义;但是当地图的紧密程度很高和 AGV的数量很多时,AGV运行时会出现大量等待延迟,这时根据具体情况提高阻塞门限,可以取得很好的效果.通过阻塞函数值与阻塞门限比较来划定搜索深度,将问题复杂度由 NP级降低为多项式级,使实时计算成为可能.

考虑可行解结构为一组路径编号 $X = [j_1, j_2, \dots, j_k]$,算法如下:

- 步骤 1:初始化:设定解 $X = [1, 1, \dots, 1]$, $k = 1$, $F(X) = \infty$, $G(X) = \infty$, 门限 $T = C$ (常量);
- 步骤 2:从左至右对第 k 层节点搜索,计算 $G(x)$ 、 $F(x)$,搜索完毕,记录下 $F(x)$ 最小的节点 x' ,

⊗ 以最先到达的 AGV的方向为准,同向的方向为正,反向的为负;

⊗ 检测冲突的时间可能性,并根据方向判定冲突类别.

如果为超前冲突,则:

如果 $F(x') < F(X)$,则 $X = x'$;

步骤 3:根据 $G(x)$ 的值扩展节点,如果 $G(x) > T$,则扩展到下层节点;否则不扩展;

步骤 4: $k++$,跳转到步骤 2,对下一层进行搜索;如果不存在下一层节点,跳转到步骤 5;

步骤 5:算法结束,即为最优解.

4.5 动态路径规划的鲁棒性

多 AGV调度系统的鲁棒性体现在对突发事件的解决上.突发事件可分两类^[8]:(1)临时突发事件;(2)永久突发事件.

对于临时突发事件,可以通过监控器,对 AGV经过节点的顺序加以控制来解决^[8].而对于永久突发事件,动态路径规划方法可以通过监控器检测的信息,接受上层控制器的重新规划的任务,对其解决.由于该动态路径规划方法可以根据实时的 AGV运行信息来动态调整 AGV的运行路径,较以往的静态的无冲突预测路径规划方法^[8]鲁棒性有很大提高.

5 仿真实例 (Simulation example)

本文通过 VC++6.0开发的仿真平台对多 AGV调度系统进行了仿真研究,仿真系统体系结构见图 3.

图 3 仿真系统体系结构

Fig. 3 Architecture of the simulation system

在该仿真系统上,对二 AGV调度在图 1所

示的地图上进行了测试.首先对 AGV1 下达任务: 8 ->15; 然后对 AGV2 下达任务 14 ->10.测试结果由图 4 给出. (a)测试 1

(b)测试 2

图 4 仿真结果

Fig. 4 Simulation results

测试 1 是在 AGV1 运行在 9 ->10 的路段时对 AGV2 实施任务触发的结果;测试 2 是在 AGV1 运行在 8 ->9 的路段时对 AGV2 实施任务触发的结果.

结果表明该策略可以很好地动态调整 AGV 的运行轨迹,实时地对运行中的多辆 AGV 实施全局优化.

6 结论 (Conclusion)

本文重在考虑现有 AGVS 研发和实施阶段存在的如研发周期长、AGV 柔性低等技术问题,提出一种两阶段动态路径规划策略.该法很好地实现了对地图信息的抽取,并离线生成 k 最短路径,避免了在线路径搜索带来的运算负担.同时,为提高 AGV 的灵活

性,本文又提出一种动态路径规划方法,实时对多个 AGV 同时规划路径,增强了系统的鲁棒性.为了满足实时性的要求,采用阻塞门限定搜索深度的启发式算法求解,将问题的复杂度降低到多项式级.通过仿真测试,证明了该策略的可行性.但是由于路径复杂度的问题,随着节点数增多和任务路径长度的增大,还是会带来实时性的问题,有待于算法的进一步改进.

参考文献 (References)

- [1] Wu N Q, Zhou M C. AGV routing for conflict resolution in AGV systems [A]. Proceedings of the 2003 IEEE International Conference on Robotics and Automation [C]. Taipei Taiwan: IEEE, 2003. 1428 - 1433.
- [2] Roszkowska E. Undirected colored Petri net for modelling and supervisory control of AGV systems [A]. Proceedings of the 6th International Workshop on Discrete Event Systems [C]. Los Alamitos, CA, USA: IEEE, 2002. 135 - 142.
- [3] 李晓萌,杨煜普,许晓鸣.基于递阶强化学习的多智能体 AGV 调度系统 [J]. 控制与决策, 2002, 17(3): 292 - 296.
- [4] Miller R D. Automated Guided Vehicles and Automated Manufacturing [M]. Dearborn, Michigan: Society of Manufacturing Engineers, 1987.
- [5] Lee J H, Lee B H, Choi M H. Real-time traffic control scheme of multiple AGV systems for collision free minimum time motion: a routing table approach [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 1998, 28(3): 347 - 358.
- [6] Deo N, Pang C Y. Shortest-path algorithms: taxonomy and annotation [J]. Networks, 1984, 14(2): 275 - 323.
- [7] Dijkstra E W. A note on two problems in connexion with graphs [J]. Numerische Mathematic, 1959, 1(2): 269 - 271.
- [8] Maza S, Castagna P. Conflict-free AGV routing in bi-directional network [A]. Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation [C]. Piscataway, NJ, USA: IEEE, 2001. 761 - 764.
- [9] Benita M B. System reliability and congestion in a material handling system [J]. Computers & Industrial Engineering, 1999, 36(3): 673 - 684.

作者简介:

刘国栋 (1979-), 男, 硕士生. 研究领域: 多 AGV 调度系统的仿真及调度策略.

曲道奎 (1961-), 男, 研究员, 博士生导师. 研究领域: 机器人学和智能控制.

张雷 (1969-), 男, 硕士生导师. 研究领域: 移动机器人规划和控制.