

基于线性时序逻辑的最优巡回路径规划

肖云涛¹ 欧林林¹ 俞立¹

摘要 基于线性时序逻辑 (Linear temporal logic, LTL) 的路径规划方法中, 多点巡回路径规划问题尚无有效解决方案. 为了在道路网络中实现最优巡回监测, 提出了基于 LTL 的最优巡回路径规划方法. 首先, 将环境建模成一个切换系统, 用 LTL 语言描述包含多个巡回点和障碍物的任务需求; 接着, 利用循环移位法构建能够融合任务需求和环境模型的扩展乘积自动机, 以建立路径信息完整的网络拓扑; 最后, 采用基于迪科斯彻法的最优综合算法搜索扩展乘积自动机网络上的最优路径, 从而获得能够满足复杂任务需求的最优巡回路径. 仿真结果表明, 该方法能够有效实现最优巡回路径规划.

关键词 线性时序逻辑, 任务需求, 切换系统, 巡回路径规划

引用格式 肖云涛, 欧林林, 俞立. 基于线性时序逻辑的最优巡回路径规划. 自动化学报, 2014, 40(10): 2126–2133

DOI 10.3724/SP.J.1004.2014.02126

Optimal Patrolling Path Planning via Linear Temporal Logic

XIAO Yun-Tao¹ OU Lin-Lin¹ YU Li¹

Abstract For a complex road network environment, the problem of patrolling path planning with multiple nodes has not be effectively solved. In order to accomplish high-level patrolling task in road network, an optimal patrolling path planning method is proposed on the basis of the theory of linear temporal logic (LTL). Firstly, the environment is modeled as a transition system and the patrolling task is described with linear temporal logic formula. Then, an extended product automaton combining the transition system and the linear temporal logic formula is constructed by implementing a circular shift algorithm such that the network topology with complete path information could be established. Finally, the Dijkstra algorithm is utilized to search the optimal path in the network of the extended product automaton, and thus, the optimal path satisfying the task requirement is correspondingly obtained. The results of the simulation experiment show the validity of the algorithm.

Key words Linear temporal logic (LTL), task requirement, transition system, patrolling path planning

Citation Xiao Yun-Tao, Ou Lin-Lin, Yu Li. Optimal patrolling path planning via linear temporal logic. *Acta Automatica Sinica*, 2014, 40(10): 2126–2133

近年来, 路径规划作为自主移动车辆、无人驾驶飞机和机器人控制等领域的基础, 已受到了越来越多关注. 传统的路径规划方法, 如人工势场法、蚁群算法和神经网络法^[1–5]等, 主要集中在规划路径使其满足“从 A 点到达 B 点并且避开障碍物”等简单的任务指令^[6], 对于实际较复杂的任务则很难进行路径规划, 例如全局保持在某些点的集合之内 (安全性), 顺序经过某几个点 (保证性) 后, 巡回其他某几个点 (循环性), 规避某些点 (避障), 到了一些点之后还必须到达另外一些点 (反应性) 等. 基于线性时

序逻辑 (Linear temporal logic, LTL) 语言能够描述上述这些复杂的任务, 因此获得了较为广泛的关注, 文献 [6–12] 提出了许多基于 LTL 的路径规划方法, 但是仍没有较好地解决对于多点巡回等更复杂的任务.

目前, 对基于 LTL 理论的路径规划研究主要集中在单机器人的巡回监测问题、多机器人的同步规划问题, 以及环境随机性、通信受限和分布式算法问题等. 其中巡回监测问题一直是这个领域研究的热点, 目前已有许多文献提出了不同的方法. 文献 [7–8] 采用最小瓶颈环法研究满足复杂任务需求的单点最优巡回路径问题, 即以最小的时间间隔上限访问所指定的某个巡回点或点集合. 它的最优性由连续两次访问所指定的巡回点或点集合的时间间隔上限来评价; 文献 [9] 基于文献 [7–8] 的方法, 将所研究的问题推广至多机器人, 并针对多个机器人之间控制精度的不同, 提出了具有鲁棒性的跟踪方法; 文献 [10–11] 也基于文献 [7–8] 的方法, 针对路径沿途回报随机变化的情况, 建立累积回报函数, 利用

收稿日期 2013-08-07 录用日期 2014-02-17
Manuscript received August 7, 2013; accepted February 17, 2014

国家自然科学基金 (61273117, 61273116), 浙江省自然科学基金 (Y1111012) 资助
Supported by National Natural Science Foundation of China (61273117, 61273116), and Natural Science Foundation of Zhejiang Province (Y1111012)

本文责任编辑 侯增广
Recommended by Associate Editor HOU Zeng-Guang

1. 浙江工业大学信息工程学院 杭州 310023
1. College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023

滚动优化的方法跟踪所得最优路径, 并使回报最大化; 文献 [12] 设计了一种基于层次法和滚动优化框架的嵌入式控制软件, 并加入反应机制, 实现了具有容错功能的路径规划。

然而, 上述文献提出的方法无法直接解决多个点的最优巡回问题. 传统的多点巡回是经典的旅行商 (Traveling salesman problem, TSP) 问题, TSP 问题的解法能够求解一条遍历网络中所有节点的较短路径, 但其局限性在于: 1) 它只存在近似解法, 无法精确获取最短巡回路径; 2) 它无法满足复杂要求, 比如规避若干节点、顺序访问若干节点后执行巡回任务等。

因此, 基于线性时序理论的多点巡回路径规划是本文所要研究的问题. 例如在道路网络中, 顺序经过某些节点后 (搜集物资), 巡回若干指定节点 (循环发放物资). 针对这个问题, 本文提出了基于循环移位法构建的扩展乘机自动机, 将 LTL 公式的周期性任务片段所转化的图表进行循环移位, 并对移位后所构成的子图表进行合并, 建立一个包含所有满足任务需求的路径可能性的网络拓扑, 然后在新构成的网络拓扑上搜索最优路径. 此方法不仅能实现多点巡回的最优路径规划, 还具有基于 LTL 规划方法的其他优点, 如实现顺序访问、规避障碍物等. 该方法为无人驾驶机空中巡回、智能汽车的自主导航、安防机器人在工厂中的巡回安全监测等方面的应用打下了良好的基础。

1 预备理论

为采用 LTL 解决路径规划问题, 首先, 要将实际环境建立成切换系统模型^[13]. 加权切换系统用一个图表 $T = (Q, q_0, \delta, \omega, \Pi, h)$ 表示, 其中 Q 是状态的有限集合, 每个状态代表道路网络中的一个节点; $q_0 \in Q$ 是初始状态, 代表初始位置; $\delta: Q \rightarrow 2^Q$ 是切换函数, 代表节点与节点之间的切换关系; ω 是权重函数, 为所有有效切换 (q_1, q_2) , $q_2 \in \delta(q_1)$ 分配一个正权值, 这个权值代表从一个节点运动到另一个节点所需的时间; Π 是原子命题的有限集合; $h: Q \rightarrow 2^\Pi$ 是状态的命题函数。

其次, 采用 LTL 语言^[14], 将任务需求以 LTL 公式的形式进行描述. 这个公式是由切换系统的状态集合 Q 或原子命题集合 Π 、标准布尔算子 \neg (非)、 \vee (或)、 \wedge (与), 以及时序逻辑算子 \bigcirc (下一步)、 \Rightarrow (直到)、 \Diamond (最后)、 \Box (始终) 所组成的表达式. 为了解决最优巡回问题, 将用于指示任务的 LTL 公式描述成如下的并列相连形式, 其中包含安全性、保证性、反应性、周期性等属性:

$$\phi = \phi_{safe} \wedge \phi_{gua} \wedge \phi_{resp} \wedge \phi_{circ} \quad (1)$$

其中,

$$\phi_{safe} = \bigvee_{j \in I_1} \Box \varphi_{1,j}$$

$$\phi_{gua} = \bigwedge_{j \in I_2} \Diamond \varphi_{2,j}$$

$$\phi_{resp} = \bigwedge_{j \in I_3} \Box (\varphi_{3,j} \Rightarrow \Diamond \psi_{3,j})$$

$$\phi_{circ} = \Box \neg \text{obs} \wedge \bigwedge_{j \in I_4} \Box \Diamond \varphi_{4,j}$$

I_1, \dots, I_4 是关于 Q 中状态的有限子集, $\varphi_{i,j}$ 和 $\psi_{i,j}$ 是集合 I_i 中的元素. 上述安全性属性确保 I_1 涉及的命题在路径执行过程中保持为真; 保证性属性确保 I_2 所涉及的命题在路径执行过程中至少一次为真; 反应性属性确保 I_3 所涉及的命题 $\varphi_{i,j}$ 在执行过程中一旦为真, 则存在后续命题 $\psi_{i,j}$ 为真; 周期性属性确保 I_4 所涉及的命题在路径执行过程中无限重复为真. 本文主要针对 LTL 的周期性属性, 将 ϕ_{circ} 中的每一个 $\varphi_{4,j}$ 作为一个周期性任务。

一条切换系统的路径 σ 是一串由状态集合 Q 中的元素所组成的状态序列. 任务公式 φ 所涉及的状态集合表示为 $[[\varphi]]$, 所涉及的状态个数表示为 $|\varphi|$. 路径 σ 满足 LTL 公式 φ 记作 $\sigma \models \varphi$.

2 问题描述

考虑一个包含多种命题节点的道路网络 (如图 1 所示). 网络中存在工厂 (物资搜集点)、住宅区 (物资发放点)、机器人充电站 (初始位置) 和障碍物区域等. 图 1 的右上角是加油站, 有烟囱的房屋为工厂区域, 无烟囱的房屋为住宅区, 灰色实线代表道路. 任务需求是类似式 (1) 形式的 LTL 公式, 用于指示如规避障碍物区域, 顺序经过某些物资搜集点, 最后在若干地点之间巡回发放物资等复杂任务. 本文不考虑搜集或发放等具体行为的实现, 只考虑规划一条能够实现上述行为的全局最优路径。

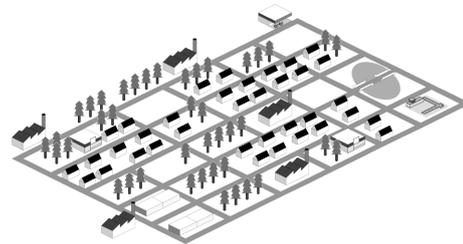


图 1 道路网络

Fig. 1 Road network

问题 1. 给定一个道路网络为式 (1) 形式的 LTL 公式, 判断是否存在路径 r , 使 $r \models \varphi$.

随后引入成本函数:

$$C(r) = \limsup_{i \rightarrow +\infty} [T_{circ}(i+1) - T_{circ}(i)] \quad (2)$$

用于标记问题 1 解的成本, 以寻求一个成本最小的最优解, 其中 $T_{circ}(i)$ 是路径 r 执行过程中第 i 次完成 ϕ_{circ} 内所有任务时的累积切换权值之和.

问题 2. 给定一个道路网络为式 (1) 形式的 LTL 公式, 判断是否存在最优路径 r^* , 使 $r^* \models \phi$, 并且对于所有 r , $C(r^*) \leq C(r)$.

在以往的研究中, 文献 [7–8] 针对一个巡回点的任务解决了问题 1; 文献 [10–11] 在文献 [7–8] 的基础上, 考虑了环境的随机性问题; 文献 [12] 针对式 (1) 形式的 LTL 公式, 设计了分层法的控制框架, 并未深入研究对于多点巡回问题. 然而多点巡回问题是著名的图论问题之一, 具有深刻的理论背景和应用价值. 虽然目前存在许多求解近似最优解的方法, 但在加入了 LTL 任务的限制条件后, 这些方法并不能直接被采用. 因此, 基于 LTL 的多点巡回问题的研究, 具有相当的挑战性.

3 基于扩展乘机自动机的最优巡回算法

3.1 环境建模与任务描述

首先, 考虑一个简单网络, 如图 2 所示的切换系统可以被看作是由实际环境经过抽样而成的网络拓扑. 状态 q_i 表示道路网络中的特定节点, 其中 q_0 是初始位置; 有向线段的权值表示从一个节点切换到另一个节点所需的成本.

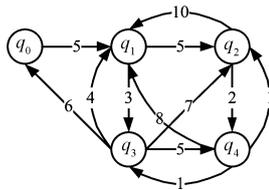


图 2 切换系统

Fig. 2 Transition system

采用式 (1) 形式的 LTL 公式描述 q_1 、 q_4 节点的巡回任务为:

$$\phi_{circ} = \square \diamond q_1 \wedge \square \diamond q_4 \quad (3)$$

这里只考虑周期性任务片段 ϕ_{circ} , 其余片段 ϕ_{gua} 和 ϕ_{resp} 可以采用传统的乘机自动机^[5] 构建图表, 并采用迪科斯彻算法^[15], 搜索初始状态至接收状态的最优路径, 以完成满足相应任务片段的路径规划, ϕ_{circ} 是无限的周期性任务. 本文将在第 3.2 节给出累积成本函数与切换系统无限任务部分相关的子图 T_{inf} , 本文在第 3.3 节构建周期性集合的扩展乘机自动机, 在第 3.4 节中描述最优巡回算法.

3.2 周期性任务集合的可到达性

在采用 LTL 理论解决路径规划问题时, 问题 1 是不可避免的, 一般只需判断初始状态是否包含在

最终位置的可达性集合内, 就能判断是否存在开始于那个节点且最终满足任务需求的路径. 为了解决问题 2, 本节将给出满足其他任务属性片段的路径相对于 ϕ_{circ} 的可达性集合, 并判断周期性任务集合内部成员之间的可达性, 同时定义了切换系统无限任务部分的子图 T_{inf} , 以缩小问题 2 的计算范围.

1) 定义切换系统 T 上某个节点的可到达性为累计成本函数 $V_T^d(s)$, 其中 s 为起点, d 为终点, 如果 d 对于 s 不可到达, 则 $V_T^d(s) = \infty$, 累计成本函数采用迪科斯彻算法实现, 见下式. 其中, $r_T^d(s)$ 为相应路径, $T.adj$ 为切换系统 T 的邻接矩阵:

$$[r_T^d(s), V_T^d(s)] = \text{dijkstra}(T.adj, s, d) \quad (4)$$

2) 构建切换系统 T 的周期性任务片段的最大子图 T_{inf} , 使其对于 ϕ_{circ} 之外所有任务片段涉及的点是可达的, 对不可到达 $[[\varphi_{4,j}]]$, $\forall j \in I_4$, 却包含在其他任务片段中的点 (LTL 任务指示具有逻辑错误), 将在算法 3 中予以排除.

算法 1. 构建 T 的最大无限部分子集 T_{inf}

输入: T , $[[\varphi_{4,j}]]$, $\forall j \in I_4$.

输出: T_{inf} .

- 1) 对于任意 $j = 1, 2, \dots, |I_4| - 1$;
- 2) 如果 $V_{\varphi_{4,j+1}}^T(\varphi_{4,j}) = \infty$;
- 3) 返回 $T_{inf} = \emptyset$;
- 4) 结束判断;
- 5) 结束循环;
- 6) 如果 $V_{\varphi_{4,1}}^T(\varphi_{4,|I_4|}) < \infty$;
- 7) 返回 $T_{inf} = \{T | V_{\varphi_{4,|I_4|}}^T(q_i) < \infty, q_i \in Q\}$;
- 8) 否则, 返回 $T_{inf} = \emptyset$;
- 9) 结束判断.

算法 1 检查了 ϕ_{circ} 集合中的点所构成的子图是否为强连通, 如果为强连通, 则返回 T 上所有能到达这个子图的点集合, 作为切换系统 T_{inf} 中的状态, 状态间的邻接关系由 T 继承而来.

3.3 扩展乘机自动机的构建

由于式 (1) 中只有 I_4 集合为周期性集合, 且其中任务都具有同样的优先级 (因为不同的优先级无关最优性, 本文不讨论具有不同优先级的情况), 任意安排集合内任务的执行顺序不影响路径对 LTL 公式的满足性. 然而, 任务的顺序关系到路径的最优性, 所以可以选择事先安排任务顺序, 或在任务执行过程中刷新任务顺序. 传统的乘机自动机事先安排了任务顺序 (即巡回点输入时的顺序), 但并非为最优顺序. 为了解决问题 2, 本文的思路是将 LTL 公式中的周期性任务片段 (巡回任务) 进行重新排列, 事先排列所有任务顺序, 以便在算法 3 中快速寻优.

采用循环移位法构建包含所有任务顺序的图表, 提出了扩展的乘机自动机:

$$P^* = \cup_{i=1}^{k!} T_{\text{inf}} \times B_i \quad (5)$$

其中, k 为 ϕ_{circ} 中巡回节点个数; $B_i = (S_{B_i}, s_0, \sum, \delta_{B_i}, F_{B_i})$ 为 Büchi 自动机, 每个 B_i 由循环移位法构建. Büchi 自动机是由 LTL 公式通过 LTL2BA 算法转化而来的图表^[16], 它用一个元组 $B = (S_B, s_0, \sum, \delta_B, F_B)$ 表示, 其中 S_B 是状态的有限集合; $s_0 \in S_B$ 是初始状态; \sum 是输入字母表; $\delta_B: S_B \times \sum \rightarrow 2^{S_B}$ 是切换函数; $F_B \subseteq S_B$ 是接收状态的集合.

假设存在一条切换系统的路径能够到达 Büchi 自动机的接收状态, 则称这条路径为 Büchi 自动机的接收路径, Büchi 自动机的接收路径即是满足任务需求的路径. 如图 3 所示, 它是将式 (3) 作为一个完整的 LTL 公式 $\phi = \square \diamond q_1 \wedge \square \diamond q_4$ 转化成的 Büchi 自动机, 而不是一个片段, 其中有向线段上的 1 代表 $[[T_\phi]]$ 中所有状态.

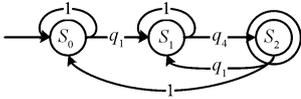


图 3 Büchi 自动机

Fig. 3 Büchi automaton

定义 1 (扩展乘机自动机). P^* 是一个元组 $(S_{P^*}, P^*_0, S_{P^*_0}, \delta_{P^*}, \omega_{P^*}, F_{P^*})$; $S_{P^*} = \cup_{i=1}^{k!} Q_{\text{inf}} \times S_{B_i}$ 是状态集合; $S_{P^*_0} = \{q_{\text{inf},0}\} \times \{s_0\}$ 是初始状态; $\delta_{P^*}: S_{P^*} \rightarrow 2^{S_{P^*}}$ 是切换函数; $\omega_{P^*}: S_{P^*} \times S_{P^*} \rightarrow \mathbf{R}^+$ 是从切换系统继承而来的正值权重函数; $F_{P^*} = \cup_{i=1}^{k!} Q_{\text{inf}} \times F_{B_i}$ 是接收状态集合.

算法 2. 构建扩展乘机自动机

输入: 切换系统 T ; (1) 形式的 LTL 公式 ϕ .

输出: 扩展乘机自动机 P^* .

- 1) 令 $k = |I_4|$;
- 2) 对于任意 $i = 1, 2, \dots, k$;
- 3) 令 $F[i] = [[\varphi_{4,i}]]$;
- 4) 结束循环;
- 5) 令 $n = k!$, $D[1] = F[1]$;
- 6) 对于任意 $i = 2, \dots, n$;
- 7) 令 $D[i] = \text{null}$;
- 8) 结束循环;
- 9) 令 $y = 2$
- 10) 当 $y < n$ 执行以下计算;
- 11) 对于任意 $i = 2, \dots, k$;
- 12) 对于任意 $j = 1, 2, \dots, n$;
- 13) 如果 $D[j] \neq \text{null}$;

- 14) 令 $D[j] = \text{strcat}(D[j], F[i])$;
- 15) 对于任意 $x = 1, 2, \dots, \text{length}(D[1]) - 1$;
- 16) 令 $\text{circshift}(D[j], 1)$;
- 17) 令 $D[y] = \text{circshift}(D[j], 1)$;
- 18) 令 $y++$;
- 19) 结束循环;
- 20) 结束判断;
- 21) 结束循环;
- 22) 结束判断;
- 23) 对于任意 $i = 1, 2, \dots, n$;
- 24) 对于任意 $j = 1, 2, \dots, k$;
- 25) 令 $\varphi_{4,j}^i = D[i](j)$;
- 26) 结束循环;
- 27) 令 $B_i = \text{LTL2BA}(\phi_{\text{circ}}^i)$;
- 28) 结束循环;
- 29) 对于任意 $i = 2, \dots, n$;
- 30) 令 $P_i = T_{\text{inf}} \times B_i$;
- 31) 结束循环;
- 32) 则 $P^* = P_1 \cup P_2 \cup \dots \cup P_n$.

图 4 是将图 2 的切换系统与式 (3) 输入算法 2 所得到的扩展乘机自动机的网络拓扑.

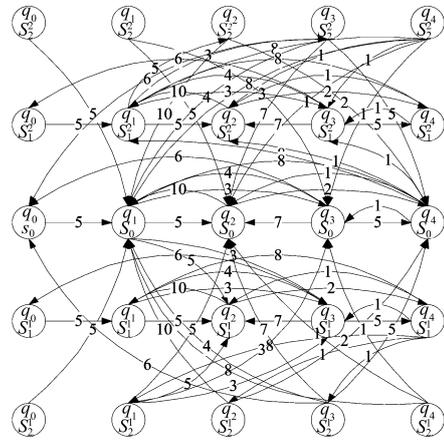


图 4 扩展乘机自动机

Fig. 4 Extended product automaton

3.4 最优巡回算法综合

为解决问题 2, 本文采用了一种分段搜索策略, 见算法 3. 思路是针对非周期性片段, 采用传统乘机自动机的方法, 搜索一条从初始状态至接收状态的最小成本路径, 乘机自动机的一条接收路径 r_P 即是存在于切换系统中且又满足 LTL 公式的路径, 这种方法已经成熟, 文中不详细说明. 针对周期性片段, 如果最优的任务顺序被包含在乘机自动机中, 则只要在其中搜索一条包含接收状态 F_P 的最小累积成本路径回环, 就能找到巡回任务片段的最优解.

本文解决巡回任务片段的方法大致分为 3 步:

1) 判断路径的当前状态 (其余片段最优路径的最后一个状态) 是否在周期性任务状态集合的可到达集合之内; 2) 以路径的当前状态为初始状态构建扩展乘机自动机; 3) 将在传统乘机自动机上搜索从初始状态至接收状态的最小成本路径算法, 切换至在扩展乘机自动机上对最优巡回路径的搜索算法. 由于巡回路径是无限重复的, 因此为了计算无限路径的成本, 在此引入路径的前缀 - 后缀格式:

定义 2 (路径的前缀和后缀形式). 扩展乘机自动机的接收路径的前缀 (*prefix*) 是一条从初始状态 S_{P^*0} 至接收状态 F_{P^*} 的状态 f 的有限序列, 但不包含 f . 路径的周期性后缀 (*suffix*) 是一条从前缀所到达的最终状态 f 出发再次到达这个状态的重复路径. 路径的形式为 $r_{P^*} = prefix_{P^*}, suffix_{P^*}, suffix_{P^*}, \dots$.

得到扩展乘机自动机上的最优路径后, 还需将其映射回初始的切换系统, 于是引入如下理论:

理论 1 (乘机自动机路径映射)^[7]. 如果 $\gamma_T(r_P) = r_T = q_0q_1 \dots$ 是一条 r_P 从乘机自动机映射到切换系统 T 上的路径, 其中 $r_P = (q_0, s_0), (q_1, s_1), \dots$ 是一条乘机自动机的可接收路径, r_T 是其映射到 T 上的路径, 且其产生的字串满足 LTL 公式 ϕ .

对于一条从初始状态出发的巡回路径, 其成本定义为按照理论 1 将扩展乘机自动机接受路径 r_{P^*} 映射回切换系统 T_{inf} 后的后缀路径长度, 如下式所示:

$$C(r^*) = \min_{r|\phi} C(r) = \min_{r|\phi} C[r(suffix)] = V_{T_{inf}}^{\varphi_{4,|I_4|}}(\varphi_{4,|I_4|}) + \sum_{i=1}^{|I_4|-1} V_{T_{inf}}^{\varphi_{4,i+1}}(\varphi_{4,i}) \quad (6)$$

算法 3. 最优巡回路径规划算法

输入: 切换系统 T , 式 (1) 类型的 LTL 公式.

输出: r^* .

- 1) $B = LTL2BA(\phi/\phi_{circ})$;
- 2) $P = T \times B$;
- 3) $[r_P^{FP}(S_{P_0}), V_P^{FP}(S_{P_0})] = dijkstra(P.adj, S_{P_0}, F_P)$;
- 4) $r_T = \gamma_T[r_P^{FP}(S_{P_0})]$;
- 5) $q_{inf,0} \leftarrow r_T$ 的最后一个状态;
- 6) 构建 T_{inf} ;
- 7) 如果 $q_{inf,0} \notin T_{inf}$;
- 8) 返回无满足要求的路径;
- 9) 结束判断;
- 10) 构建 P^* ;
- 11) $M_{P^*} = \{(q_i, s_k) \in S_{P^*}/F_{P^*}\}$;
- 12) 对于任意 $s_{p^*} \in F_{P^*}$;

- 13) $[suf_1, L_1] = dijkstra(P^*.adj, s_{P^*}, M_{P^*})$;
- 14) 对于任意 $s_m \in M_{P^*}$;
- 15) $[suf_2, L_2] = dijkstra(P^*.adj, s_m, s_{P^*})$;
- 16) 结束循环;
- 17) $L = L_1 + L_2$, 取出 L 中最小值对应的 suf_1 与 suf_2 ;
- 18) 结束循环;
- 19) 删除 suf_1 和 suf_2 的最后一位;
- 20) $suffix_{P^*} = suf_1 + suf_2$;
- 21) $[prefix_{P^*}, L_P] = dijkstra(P^*.adj, S_{P^*0}, suffix_{P^*})$;
- 22) 取出 L_P 中最小值对应的 $prefix_{P^*}$;
- 23) 删除 $prefix_{P^*}$ 的最后一位;
- 24) 循环左移 $suffix_{P^*}$ 序列, 使其中对应 $prefix_{P^*}$ 被删除的状态为序列的开头;
- 25) $r_{P^*} = prefix_{P^*}, suffix_{P^*}, suffix_{P^*}, \dots$;
- 26) $r_{T_{inf}} = \gamma_{T_{inf}}(r_{P^*})$;
- 27) 删除 r_T 最后一位;
- 28) 返回 $r^*(prefix) = [r_T, r_{T_{inf}}(prefix_{P^*})]$;
- 29) $r^*(suffix) = r_{T_{inf}}(suffix_{P^*})$;
- 30) 返回 $r^* = [r^*(prefix), r^*(suffix), r^*(suffix), \dots]$.

图 5 是采用算法 3 在图 4 所示的扩展乘机自动机上搜索到的最优巡回路径: $r_{P^*} = (q_0, s_0), (q_1, s_0), (q_2, s_1^1), (q_4, s_1^1), (q_3, s_2^1), (q_1, s_2^1), \dots$, 图 6 是映射回切换系统的最优巡回路径: $r^* = q_0q_1q_2q_4q_3, q_3q_1q_2q_4, q_3q_1q_2q_4, \dots$, 其中细线代表路径前缀, 粗线所围成的回路代表路径后缀.

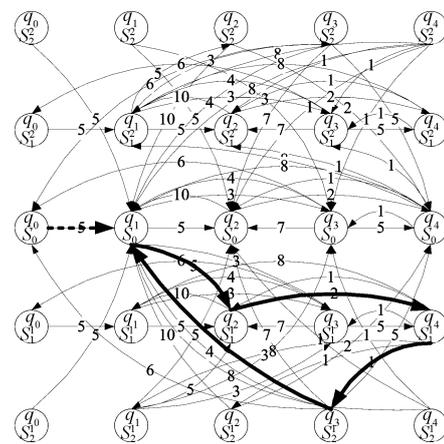


图 5 扩展乘机自动机上的路径

Fig. 5 Path on extended product automaton

上述巡回算法的最优性由以下两点保证: 1) 算法 2 使扩展乘机自动机中包含了所有巡回点的访问顺序; 2) 算法 3 底层采用迪科斯彻方法, 确保具体两点间路径最短, 然后通过各种路径的组合, 比较其

成本之和的大小, 确保返回的巡回环路是成本最小的, 因此最优性得以保证.

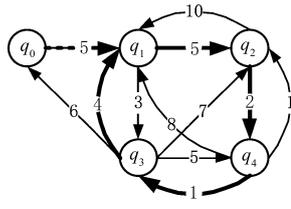


图 6 切换系统上的路径
Fig. 6 Path on transition system

4 仿真算例

假设一个包含多种命题节点的道路网络, 其中包括障碍物与巡回节点. 例如图 1 所示的道路网络. 假设交通规则规定机器人或无人驾驶汽车可以行驶在道路上, 但禁止穿越住宅区.

首先, 根据图 1 建立切换系统模型, 如图 7 所示. 把每一条横向路段作为切换系统的状态, 假设每条路段长度为 1, 加油站所在区域为初始位置 q_0 ; 工厂所在区域 $q_{12}, q_{15}, q_{23}, q_{36}, q_{40}$; 住宅区所在路段为障碍物 $q_i \in obs, i \in \{8, 13, 14, 16, 17, 22, 25, 26, 31, 32\}$, 见图中黑色状态, 可行路段由白色状态表示.

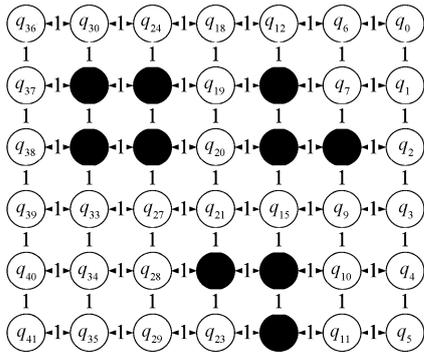


图 7 道路网络的切换系统模型
Fig. 7 Transition system model of road network

例 1. 任务需求为巡回监测工厂.

图 1 存在 5 个工厂, 因此这个任务可以看作一个典型的多点巡回任务, 且巡回点数 $|I_4| = 5$. 那么, 描述其任务需求的 LTL 公式可以写成:

$$\phi_1 = \phi_{circ} = \square \neg obs \wedge \square \diamond q_{12} \wedge \square \diamond q_{15} \wedge \square \diamond q_{23} \wedge \square \diamond q_{36} \wedge \square \diamond q_{40}$$

从图 7 可以看出, 从 q_0 出发, 满足任务需求的路径可能为 $r: q_0 \rightarrow (q_{12} \rightarrow q_{40} \rightarrow q_{23} \rightarrow q_{15} \rightarrow q_{36})$, 其中成本 (后缀成本) 为 $C(r) = 31$; 或者是 $r: q_0 \rightarrow (q_{15} \rightarrow q_{40} \rightarrow q_{23} \rightarrow q_{36} \rightarrow q_{12})$, 其中成本 (后

缀成本) 为 $C(r) = 27$. 其中, 箭头表示从目前状态至后继状态的最短路径, 例如 $q_0 \rightarrow q_{12}$ 表示 q_0, q_6, q_{12} 这条路径; 括号中的部分表示无限循环的路径, 例如 $(q_{12} \rightarrow q_{40} \rightarrow q_{23} \rightarrow q_{15} \rightarrow q_{36})$ 表示依次由最优路径通过 $q_{12}, q_{40}, q_{23}, q_{15}, q_{36}$, 并最终由 q_{36} 以最短路径回到 q_{12} .

为方便描述, 现暂且将构建传统乘机自动机并搜索接收路径的方法^[17] 称为传统方法. 为比较本文方法与传统方法的结果, 将 ϕ_1 分别输入传统方法与本文的算法 3, 所得到的路径如图 8 和图 9 所示, 其中实线表示无限巡回路径, 虚线表示从初始状态到达无限巡回环路的路径.

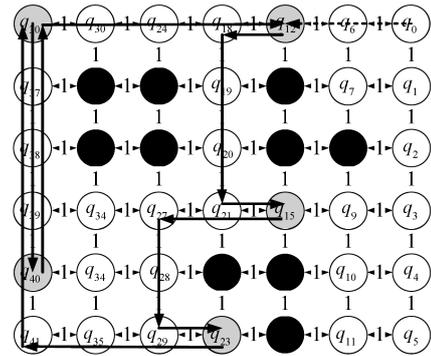


图 8 ϕ_1 经传统方法所得的路径 r
Fig. 8 Path of ϕ_1 under traditional method

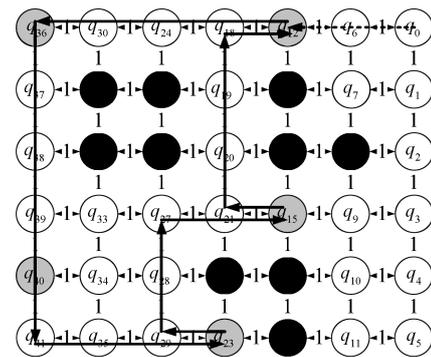


图 9 ϕ_1 经算法 3 所得的路径 r^*
Fig. 9 Path of ϕ_1 under Algorithm 3

由图 8 和图 9 可以看出, 传统方法所得的路径的无限巡回部分在 q_{36} 与 q_{40} 之间往返了多次, 这其实是不必要的, 因为每次巡回只要经过巡回点一次就能满足任务的需求; 而算法 3 优化了这一结果, 以最优的顺序经过这些点正好一次, 在满足任务需求的同时, 达到了路径的最优. 它们具体的路径以及成本如下所示:

$$r: q_0 \rightarrow (q_{12} \rightarrow q_{15} \rightarrow q_{23} \rightarrow q_{36} \rightarrow q_{40}), C(r) = 30$$

$$r^*: q_0 \rightarrow (q_{12} \rightarrow q_{36} \rightarrow q_{40} \rightarrow q_{23} \rightarrow q_{15}),$$

$$C(r^*) = 22$$

例 2. 任务为首先访问地图正北面的工厂, 然后访问西北面的工厂, 最后巡回监测其余工厂.

这个任务要求顺序访问 2 个点之后, 巡回其余 3 个点, 因此是一个典型的保证性任务 ϕ_{gua} 和巡回任务 ϕ_{circ} 的并列相连形式, 且巡回点数 $|I_4| = 3$. 采用 LTL 语言描述任务需求:

$$\begin{aligned} \phi_2 &= \phi_{gua} \wedge \phi_{circ} = \diamond q_{12} \wedge \diamond q_{36} \wedge \\ &\square \neg obs \wedge \square \diamond q_{23} \wedge \square \diamond q_{40} \wedge \square \diamond q_{15} \end{aligned}$$

在进行路径规划前, 我们能够预见, 巡回路径是唯一的, 因为 3 个节点构成的回路是唯一的. 但是由于路径访问巡回点顺序的不同, 由初始状态进入巡回回路的路径就可能不同. 因此, 将公式 ϕ_2 输入传统方法与本文的算法 3, 所得到路径如图 10 和图 11 所示.

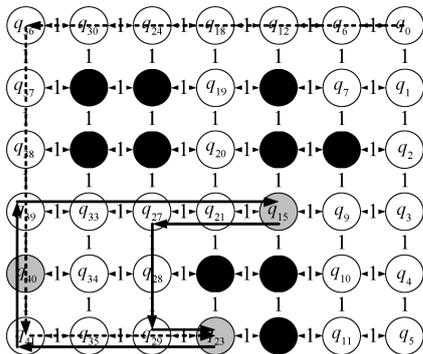


图 10 ϕ_2 经传统方法所得的路径 r

Fig. 10 Path of ϕ_2 under traditional method

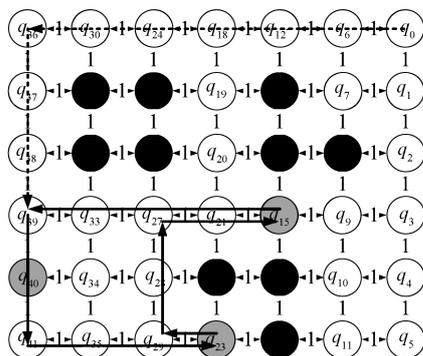


图 11 ϕ_2 经算法 3 所得的路径 r^*

Fig. 11 Path of ϕ_2 under Algorithm 3

由图 10 和图 11 可以看出, 虽然这 2 种路径的无限巡回部分的成本相等, 但算法 3 所得路径, 其从初始位置进入无限巡回部分的路径是最短的 (由于必须首先访问 q_{12} 随后再访问 q_{36} 才能开始巡回, 因此可以认为巡回任务片段是以 q_{36} 为初始位置的), 而传统方法只是按照任务输入的顺序访问, 没有事

先安排最优顺序, 所以它执行完保证性任务之后的路径从 q_{36} 开始首先到达了 q_{23} , 然后折返, 开始巡回路径. 而算法 3 所得路径直接从 q_{39} 这个最优优点进入了巡回路径. 它们具体的路径以及成本如下所示:

$$\begin{aligned} r: & q_0 \rightarrow q_{12} \rightarrow q_{36} \rightarrow (q_{40} \rightarrow q_{23} \rightarrow q_{15}), \\ & C(r) = 14, C[r(prefix)] = 14 \\ r^*: & q_0 \rightarrow q_{12} \rightarrow q_{36} \rightarrow (q_{40} \rightarrow q_{23} \rightarrow q_{15}), \\ & C(r^*) = 14, C[r^*(prefix)] = 9 \end{aligned}$$

5 总结

本文采用基于 LTL 的路径规划方法, 针对任务中包含多个巡回节点的情况, 提出了基于循环移位法构建的扩展乘机自动机, 并在扩展乘机自动机网络上采用基于迪科斯彻法的最优巡回路径规划算法搜索最优路径. 仿真研究与实验结果表明, 该方法能有效实现在满足任务需求基础上的最优巡回路径规划.

关于 LTL 的多点巡回问题, 将来有多个研究方向: 1) 为了构建更完整的拓扑网络, 本文采用了循环移位法, 虽然事先缩减了拓扑规模, 但在巡回点很多的情况下, 计算量仍然较大, 因此将来工作可以致力于减小算法复杂度上; 2) 根据所得到的最优路径, 如何针对机器人或无人驾驶汽车的模型进行控制器设计, 以实现路径的跟踪, 也是一个非常有意义的研究课题.

References

- Zhu Yi, Zhang-Tao, Song Jing-Yan. Study on the local minima problem of path planning using potential field method in unknown environments. *Acta Automatica Sinica*, 2010, **36**(8): 1122–1130
(朱毅, 张涛, 宋靖雁. 未知环境下势场法路径规划的局部极小问题研究. *自动化学报*, 2010, **36**(8): 1122–1130)
- Zhu Qing-Bao. Ant algorithm for path planning of mobile robot in a complex environment. *Acta Automatica Sinica*, 2006, **32**(4): 586–593
(朱庆保. 复杂环境下的机器人路径规划蚂蚁算法. *自动化学报*, 2006, **32**(4): 586–593)
- Fan Chang-Hong, Chen Wei-Dong, Xi Yu-Geng. A neural networks-based approach to safe path planning of mobile robot in unknown environment. *Acta Automatica Sinica*, 2004, **30**(6): 816–823
(樊长虹, 陈卫东, 席裕庚. 未知环境下移动机器人安全路径规划的一种神经网络方法. *自动化学报*, 2004, **30**(6): 816–823)
- Khatib O. Real-time obstacle avoidance for manipulator and mobile robots. *The International Journal of Robotics Research*, 1986, **5**(1): 90–98
- Yuan Kui, Li Yuan, Fang Li-Xin. Multiple mobile robot systems: a survey of recent work. *Acta Automatica Sinica*, 2007, **33**(8): 785–794
(原魁, 李园, 房立新. 多移动机器人系统研究发展近况. *自动化学报*, 2007, **33**(8): 785–794)

- 6 Kloetzer M, Belta C. Automatic deployment of distributed teams of robots from temporal logic motion specifications. *IEEE Transactions on Robotics*, 2010, **26**(1): 48–61
- 7 Smith S, Tumova J, Belta C, Rus D. Optimal path planning for surveillance with temporal logic constraints. *The International Journal of Robotics Research*, 2011, **30**(14): 1695–1708
- 8 Smith S, Tumova J, Belta C, Rus D. Optimal path planning under temporal logic constraints. In: Proceedings of International Conference on Intelligent Robots and Systems. Taipei, China: IEEE, 2010. 3288–3293
- 9 Ulusoy A, Smith S, Xu Chu-Ding, Belta C, Rus D. Optimality and robustness in multi-Robot path planning with temporal logic constraints. *The International Journal of Robotics Research*, 2013, **32**(8): 889–911
- 10 Xu C D, Lazar M, Belta C. Receding horizon temporal logic control for finite deterministic systems. In: Proceedings of American Control Conference. Montreal, Canada: IEEE, 2012. 715–720
- 11 Wongpiromsarn T, Ufuk T, Murray R. Receding horizon temporal logic planning for dynamical systems. In: Proceedings of 48th IEEE Conference on Decision and Control. Shanghai, China: IEEE, 2009. 5997–6004
- 12 Wongpiromsarn T, Ufuk T, Murray R. Receding horizon temporal logic planning. *IEEE Transactions on Automatic Control*, 2012, **57**(11): 2817–2830
- 13 Belta C, Isler V, Pappas G. Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics*, 2005, **21**(5): 864–874
- 14 Clarke E, Grumberg O, Peled D. *Model Checking*. Massachusetts: MIT Press, 1999. 27–46
- 15 Dijkstra E. A note on two problems in connexion with graphs. *Numerical Mathematic*, 1959, **1**(1): 269–271
- 16 Gastin P, Oddoux D. Fast LTL to Büchi automata translation. In: Proceedings of the 13th International Conference on Computer Aided Verification. Paris, France: IEEE, 2001. 53–65
- 17 Kloetzer M, Belta C. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 2008, **53**(1): 287–297



肖云涛 浙江工业大学信息工程学院硕士研究生. 主要研究方向为机器人路径规划, 嵌入式系统.

E-mail: ctcexiaoyt@foxmail.com

(**XIAO Yun-Tao** Master student at the College of Information Engineering, Zhejiang University of Technology. His research interest covers robot path planning and embedded system.)



欧林林 浙江工业大学信息工程学院副教授. 主要研究方向为时滞系统, 多智能体协同控制系统. 本文通信作者.

E-mail: linlinou@zjut.edu.cn

(**OU Lin-Lin** Associate professor at the College of Information Engineering, Zhejiang University of Technology. Her research interest covers theoretical aspects of time-delayed systems and control, and cooperative control of multi-agent system. Corresponding author of this paper.)



俞立 浙江工业大学信息工程学院教授. 主要研究方向为鲁棒控制, 时滞系统, 分布式控制, 网络控制系统.

E-mail: lyu@zjut.edu.cn

(**YU Li** Professor at the College of Information Engineering, Zhejiang University of Technology. His research interest covers robust control, time-delay systems, decentralized control, and networked control systems.)