

# 基于线性时序逻辑的移动端快递派送路径规划<sup>①</sup>

禹鑫焱<sup>②</sup> 郭永奎 欧林林<sup>③</sup> 汪 涛 卢 靓 张爱美

(浙江工业大学信息工程学院 杭州 310000)

**摘要** 研究了快递派送的路径规划。针对目前快递派送任务点较多,快递员不熟悉派送区域的问题,提出了一种基于线性时序逻辑(LTL)的移动端多点快递派送路径规划方法。该方法利用移动端的百度地图应用包实现快递员的定位与导航任务,提出两层(顶层、底层)规划策略。顶层规划避开百度地图实际道路环境繁琐又庞大的建模,将快递派送问题转化为旅行推销员问题(TSP),仅将快递员派送任务地点建模为一个有限状态的加权切换系统,状态之间的切换权重基于百度地图的自驾导航距离,而非简单的直线距离,以达到底层规划结果符合实际环境,确保最后搜索的路径最优性,同时利用线性时序逻辑语言描述多点快递派送任务,从而将切换系统信息与派送任务信息相融合,构建一个任务可行网络拓扑,并在该网络拓扑上基于 Dijkstra 算法搜索出快递员离散的最优路径。底层规划完成离散路径的连续化,离散路径的任意相邻任务点间的路径规划基于百度地图实现,从而实现顶层规划的离散路径连续化,获得快递员可派单的实际派送路线。实验结果表明该方法能够解决多点派送任务与派送区域受约束的问题,并保证快递员派单路径的最优性。

**关键词** 快递派送, 路径规划, 线性时序逻辑(LTL), 移动端

## 0 引言

近年来,互联网得到迅速发展,衍生的电子商务也越来越普遍,网购也被大多数人所接受,随之带来的是快递物流压力,传统的快递物流模式已无法满足现在的生产生活需求。在物流的仓储拣货阶段,传统的人工拣货已逐渐被仓储机器人拣货代替,智能化仓库已经被大量研究,亚马逊公司装备 Kiva 机器人的智能化仓库已经带来巨大效益。而快递派送是物流的末端环节,其时效性直接影响客户的满意度,派送环节就是检验快递服务水平和质量的关键<sup>[1]</sup>。目前,快递员派送的区域受到快递员对该区域熟悉度的限制,此外,派送点较多时,快递员难以

凭借对地理熟悉度规划一条最优派送路线,而现有的地图导航功能实现的也仅仅是两点之间的导航,这些约束严重影响了快递派送的时效性。根据快递员派送任务规划一条最优路径,使派送距离最短,将会提高快递员的派送效率,多点快递派送任务的路径规划研究具有重要的理论与现实意义。

快递员派送任务的路径规划问题,也就是快递员从货物分发处出发,遍历所有派送点,最后回到分发处,找到一条最短的路径,该问题属于典型的旅行推销员问题(travelling salesman problem, TSP)。当前,关于路径规划问题,已经有了一些研究成果。文献[2]基于 Dijkstra 算法研究了快递车辆配送路径优化问题,考虑了配送车辆行驶的最大距离以及配送负载约束建立约束条件,从配送起点开始搜索,依

<sup>①</sup> 863 计划(2014AA041601-05),浙江省公益项目(2016C31064)和浙江省自然科学基金(LY15F030015)资助项目。

<sup>②</sup> 男,1979 年生,博士;研究方向:机器人控制与规划;E-mail: yuxinyinet@163.com

<sup>③</sup> 通讯作者,E-mail: linlinou@zjut.edu.cn

(收稿日期:2017-03-17)

次搜索距离当前节点最近的地点作为下一节点,直至遍历所有派送点;文献[3]将 A\* 算法与贪心算法相结合来解决物流配送路径的选择,A\* 算法加入了启发函数可以提高搜索效率,利用贪心算法思想实现多点派送的目的。但以上这两种方法,都是在经典的针对点对点路径规划算法基础之上,结合贪心算法思想实现多点派送的路径规划任务,该方法很难获得全局最优路径,大多得到的是次优路径,Dijkstra 算法能保证两点间路径的最优性,但是无法获得多点快递派送这类复杂任务的最优路径。

针对复杂的任务,比如快递派送的遍历任务、持续监控工厂几个任务点的巡回任务,随着模态逻辑的深入研究,时序逻辑语言,例如计算树逻辑(CTL)和线性时序逻辑(LTL)<sup>[4]</sup>能够便捷地描述复杂的时序任务,时序逻辑理论在路径规划领域的应用越来越被关注<sup>[5-7]</sup>。文献[8]提供了一种方法解决了一类具有时序任务约束的机器人运动规划,文中采用三角分割法对环境进行离散抽象建模,采用 LTL 公式描述机器人按序访问、巡回等复杂的任务需求,用模型检测工具 NUSMV<sup>[9]</sup> 规划出满足任务的离散路径,最后设计控制规则将离散路径转化为连续路径。文献[10, 11]结合 Büchi 自动机理论,考虑时间权重,设计了最小瓶颈算法搜索出一条耗时最短的最优路径,解决了多点巡回问题。但该方法受任务节点顺序的影响,使得搜索路径并非全局最优,针对此问题,文献[12]设计了一种扩展乘积自动机方法,确保多点巡回问题,不受任务公式顺序影响,所得路径为全局最优。文献[13]根据仓储环境,设计了灵活可扩展的加权切换系统,将线性时序逻辑应用到仓储环境的移动机器人路径规划领域,解决仓储机器人拣货最优路径问题。

文献[14]以派送员经过的总路程最短作为目标函数建立数学模型,基于最小生成树的深度优先搜索算法,搜索出满足任务的“亚优解”。文献[15]对粒子群算法进行改进,提出交换操作数与交换序列理念,解决了 TSP 问题。文献[16]将粒子群算法与百度基于位置的服务(location based service, LBS)技术相结合,利用 LBS 技术实现派送人员定位任务,并利用粒子群算法搜索可行派送路线。文

献[17]采用最优化理论构建了商业区快递派送路径规划的模型,最后将蚁群算法应用于快递路径选择。文献[18]提出了一种基于自动快递机的快递派送车辆路径优化方法,该方法以配送车辆最少和车辆总行驶路程最短为目标建立数学规划模型,并考虑了遗传算法的不足,采用元胞遗传算法对建立的模型求解。但上述方法大多采用的是智能搜索算法,搜索所得路径是次优解,并不能保证是路径最短,而目前基于线性时序逻辑理论的路径规划方法可以满足复杂任务获得最优路径,但难以适用于快递员不断变化的派送环境。在文献[13, 16]的启发下,本文提出了一种基于线性时序逻辑的移动端快递派送路径规划方法。

本文基于线性时序逻辑理论,根据快递员派送任务,利用百度地图开发包创建切实可行的有限状态的加权切换系统,并在任务可行网络拓扑上利用 Dijkstra 算法搜索出最优的离散路径,获取快递派送顺序,最后基于 Android 操作系统,利用百度地图开发包进行快递员的定位与导航,将离散的路径转换为连续的实际派送路线。本方法可以解决由于快递员对派送区域熟悉度受限的问题,扩大派送范围,并能够一次性派送多点且保证派送路径最优,而非次优路径,同时,当派送过程中遇到道路拥堵时,可以根据百度地图的实时导航做局部优化,从而实现互联网+的智慧派送,提高快递员的派送效率,满足当前的快递派送需求。

## 1 预备理论

### 1.1 有限状态加权切换系统

基于线性时序逻辑理论的路径规划,需要考虑构建一个加权的有限状态切换系统(weighted finite-state transition system, WFTS)。WFTS 是一个元组  $T = (Q, q_0, \delta_T, AP, L_T, \omega_T)$ , 其中  $Q$  是一个有限的状态集;  $q_0 \in Q$  是初始状态,代表派送员派送的起点;  $\delta_T \subseteq Q \times Q$  代表切换关系;  $AP$  代表原子命题集;  $L_T: Q \rightarrow 2^{AP}$  代表标识函数集;  $\omega_T: \mathbb{R} \rightarrow \mathbb{R}_{>0}$  代表两状态之间切换的成本(时间,距离等)。

在切换系统中,  $L_T(q)$  表示状态  $q$  处的原子命

题成立。当  $q_2$  为  $q_1$  的后续状态时,  $q_2 \in \delta(q_1)$  成立, 以时间权重为例, 当系统在时间  $t_1$  时, 为状态  $q_1$ , 那么在时间  $[t_1 + \omega_T(q_1, q_2)]$  时, 系统处于状态  $q_2$ 。在加权切换系统中的一条可运行路径  $r_r = q_0 q_1 q_2 \cdots q_i$ , 其中  $q_i \in Q$ , 且对于任意的  $i \geq 0$  都有  $q_{i+1} \in \delta(q_i)$  成立。

## 1.2 线性时序逻辑语言 (LTL)

针对快递员派送任务, 线性时序逻辑语言可以方便地描述这些任务, 它由原子命题和操作符构成, 具有如下形式:

$$\phi ::= T \mid \alpha \mid F\phi_0 \mid \phi_1 \vee \phi_2 \mid G\neg \phi_3 \mid \phi_4 U\phi_5$$

其中,  $\alpha \in AP$  是一个原子命题, 符号  $\vee$  (或) 和  $\neg$  (非) 是标准布尔操作符,  $F$  (最终),  $G$  (总是) 和  $U$  (直到) 是时序操作符,  $F\phi_0$  表示  $\phi_0$  的最终状态为真, 实现访问,  $G\neg \phi_3$  表示全局总是避免  $\phi_3$ , 可用于避障,  $\phi_4 U\phi_5$  表示直到  $\phi_5$  为真,  $\phi_4$  一直保持为真。

## 1.3 Büchi 自动机 (Büchi Automaton)

线性时序逻辑公式和 Büchi 自动机有对应关系, 通过 LTL2BA 工具包<sup>[19]</sup>可将线性时序逻辑公式转化为一个 Büchi 自动机。Büchi 自动机是一个元组  $A_\phi ::= (S_\phi, S_0, \sum_\phi, \delta_\phi, F_\phi)$ , 其中  $S_\phi$  代表一个有限的状态集;  $S_0 \in S_\phi$  代表初始状态;  $\sum_\phi$  代表输入的字符表;  $\delta_\phi \subseteq S_\phi \times \sum_\phi \times S_\phi$  代表切换函数;  $F_\phi \subseteq S_\phi$  代表最终状态集。当 Büchi 自动机的一个输入为  $\omega = L(q_0)L(q_1)L(q_2)\cdots L(q_i)$ , 会对应生成一个 Büchi 自动机运行序列  $r_\phi = s_0 s_1 s_2 \cdots s_i$ , 其中  $(s_i, \omega_i, s_{i+1}) \in \delta_\phi, i \in \mathbb{N}$ 。

线性时序逻辑是一种模态逻辑, 给定一个有限状态切换系统, 需要检测该切换系统是否可以满足该时序逻辑即本文的时序派送任务公式, Büchi 自动机就是一种检测工具, 通过 LTL 任务公式对应的 Büchi 自动机可以判定该切换系统是否存在满足该任务的路径。文献[19]则提出了一种 LTL 任务公式快速转换为 Büchi 自动机的算法, 并提供了 LTL2BA 工具包 java 接口, 该接口能够实现输入给定 LTL 任务公式, 输出相应的 Büchi 自动机。

## 2 问题描述

订单派送是物流系统的重要环节, 订单派送效

率直接影响消费者的购物体验。快递员需要根据配送任务寻找出一条最优路径来缩短派送时间, 提高订单配送效率。目前快递员配送有两大瓶颈: 派送区域限制与一次派单数量限制。

如今快递员派送订单的范围都是局限在快递员熟悉的一片区域, 主要依靠对地理环境的熟悉程度来派送订单, 然而快递员对地理环境的记忆能力是有限, 这就导致了派送区域限制, 如果增加快递分发点必然会增加投入成本。同时, 在快递员一次派送货物承重一定的前提下, 针对多派送点任务, 快递员很难自主规划出多点派送的最优路径, 多走的路程严重影响派送效率, 如果快递员减少一次派件数量, 需要频繁往返快递分发处, 同样影响快递派送的时效性。

本文需要解决的问题是: 在快递员派送承重最大的基础之上, 给定快递员多点的派送任务, 并且快递员的派送地点不受快递员记忆限制, 可以是快递员熟悉的地点也可以是快递员从未去过的地点, 规划出最优派送路线, 提高快递员派送效率。

## 3 基于 LTL 的移动端快递派送路径规划

本文提出的基于线性时序逻辑理论的移动端快递派送路径规划方法, 在 Android 系统的智能手机平台上, 利用百度地图开发包构建符合实际派送环境的加权切换系统, 同时利用线性时序逻辑语言描述多点派送任务, 并将其转化为 Büchi 自动机, 接着将切换系统与派送任务相融合, 构建任务可行网络拓扑, 在任务可行网络拓扑上利用 Dijkstra 算法搜索最优路径, 然后将该路径映射回百度地图, 获得离散的最优路径, 最后利用百度地图开发包的定位导航功能, 实现离散路径的连续化。其流程图如图 1 所示。

### 3.1 环境建模

快递派送涉及的地理区域较大, 如果仅依靠派送员的区域熟悉度来划定派送区域, 必定会导致派送区域小的问题, 设计合适的环境建模方法是保证最后规划出来的路径最优, 满足派送任务需求的关键。文献[13]采用分层思想, 选取仓储环境中固

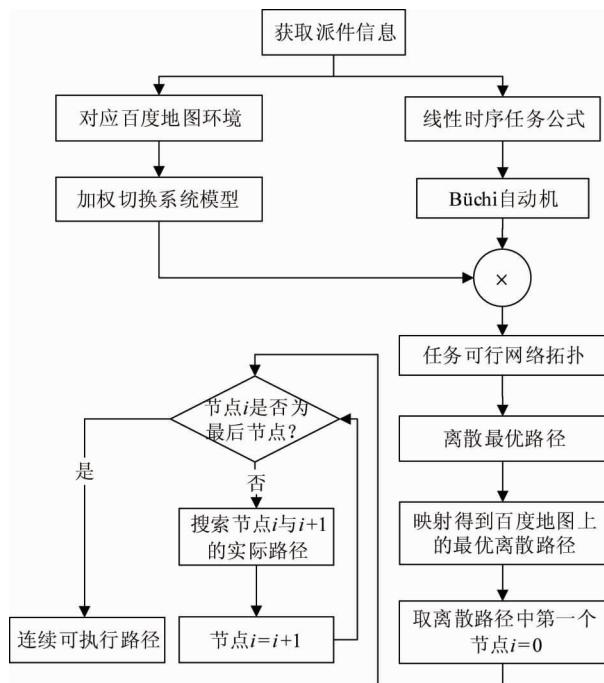


图1 基于LTL的移动端快递派送路径规划

定的路径节点构建切换系统，并没有将所有货架进行建模，接着根据取货任务货架，对切换系统进行扩展，该方法适用于规模相当的仓储系统，但快递派送的区域太大，而且实际道路网络交叉太多，选取道路交叉口作为路径节点，会导致切换系统过于庞大，从而使在任务可行网络拓扑搜索最优路径的搜索效率被严重影响，无法适用于快递实际派送任务。

随着智能手机的普遍使用，本文在Android平台上，利用百度地图开发包进行导航与加权切换系统的构建。百度图的导航能够使派送员在不熟悉该区域完成派送任务，并且百度地图开发包两点之间的导航提供了步行、公交和自驾三种模式。显然，在实际派送过程，派送员的出行方式是自驾方式，而非步行与公交模式，所以本文根据派送任务点，将自驾导航获取的任务点之间的行驶距离作为它们间的切换权重，距离是通过调用百度地图开发包的两点驾驶距离函数得到，即算法一中的 $BmapDrivingDis()$ ，进而将任务点构建为有限状态的加权切换系统WFTS，避免了道路交叉口建模造成的WFTS过于庞大问题，并且选取自驾导航的花费作为切换权重更加符合快递派送实际环境，具体过程如算法1：

算法1：构建加权切换系统  $T(\text{Construct}T())$

输入：派送起点  $P_0$ ，派送地点集  $P$

输出：加权切换系统  $T$

- (1)  $Q = P$
- (2)  $q_0 = P_0$
- (3) 对于任意的  $i = 0, 1, 2, \dots, n$
- (4) 对于任意的  $j = 0, 1, 2, \dots, n$
- (5) 如果  $i \neq j$
- (6)  $q_j \in \delta(q_i)$ ,
- (7)  $\omega(q_i, q_j) = BmapDrivingDis(q_i, q_j)$
- (8) 否则  $q_j \notin \delta(q_i)$ ,  $\omega(q_i, q_j) = \inf$
- (9) 结束判断
- (10) 结束循环
- (11) 结束循环
- (12) 返回  $T$

若快递员的派送任务为“从快递分发处  $P_0$  出发，到  $P_1, P_2$  和  $P_3$  三个地点派送快递，然后返回快递分发处  $P_0$ ”，根据算法1可得到对应的加权切换系统WFTS，如图2所示，图中箭头代表切换关系，数字代表切换权重，可用一个邻接矩阵  $T.\text{adj}$  进行表示。

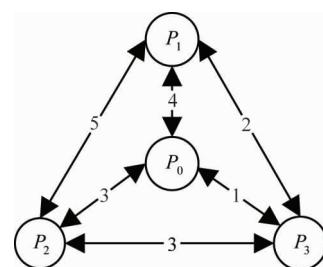


图2 加权切换系统

### 3.2 任务描述

线性时序逻辑是一种模态逻辑，它可以方便描述一些复杂的时序任务，工厂数据监控与上传的巡回任务，多任务点遍历任务以及按序访问任务都可以利用线性时序逻辑进行描述。针对3.1的任务描述，“从快递分发处  $P_0$  出发，到  $P_1, P_2$  和  $P_3$  三个地点派送快递，然后返回快递分发处  $P_0$ ”，令  $P_0$  为搜索起点，即  $T. q_0 = P_0$ ，线性时序逻辑语言描述如下：

$$F p_1 \wedge F p_2 \wedge F p_3 \wedge GF p_0 \quad (1)$$

由于任务公式(1)无法利用现有理论直接与构

建的加权切换系统相结合,任务信息与环境信息不能相融合,最后搜索路径难以同时满足任务信息与环境信息,本文利用 Büchi 自动机理论,通过 LTL2BA 工具包将线性时序逻辑公式转化为 Büchi 自动机,Büchi 自动机具有图表形式,式(1)对应的 Büchi 自动机如图 3 所示,图中共有 9 个状态,init 代表初始状态,节点 8 代表最终接收状态。

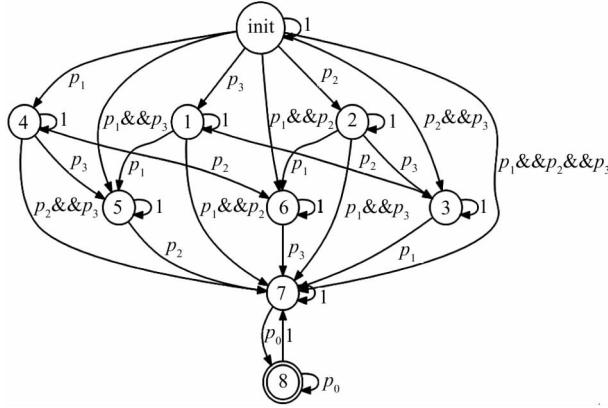


图 3 式(1)对应 Büchi 自动机

### 3.3 路径搜索

根据快递员派送任务,获得相应的加权切换系统  $T$  与任务公式对应的 Büchi 自动机后,需要将环境信息与任务信息相融合以确保最终搜索的路径既满足环境信息又符合快递派送需求,本文通过将加权切换系统与 Büchi 自动机笛卡尔乘积,构建任务可行网络拓扑(Product 自动机),即  $A_P = T \otimes A_\phi$ 。它也是一个元组  $A_P = (S_P, S_{P0}, \delta_P, \omega_P, F_P)$ , 其中  $S_P$

$= Q \times S_\phi = \{(q, s) \in S_P \mid \forall s \in S_\phi, \forall q \in Q\}$  是 Product 自动机的状态集;  $S_{P0} = \{q_0\} \times S_0$  代表初始状态集;  $\delta_P: S_P \rightarrow 2^{S_P}$  代表状态间的切换函数,其定义为当且仅当  $q_j \in \delta_T(q_i)$  并且  $s_l \in \delta_\phi(s_k, L_B(q_i))$  时,  $(q_j, s_l) \in \delta_P((q_i, s_k))$  成立;  $\omega_P: S_P \times S_P \rightarrow R^+$  继承自  $T$  且为正的权重函数,即当  $(q_j, s_l) \in \delta_P((q_i, s_k))$  时,则  $\omega_P((q_i, s_k), (q_j, s_l)) = \omega_T(q_i, q_j)$ ;  $F_P = Q \times F_\phi$  代表一个最终的接收状态集。对于任务可行网络拓扑的一个搜索路径  $r_P$ ,如果  $\text{inf}(r_P) \cap F_P \neq \emptyset$ ,那么此  $r_P$  是可被接受的,其中  $\text{inf}(r_P)$  代表路径的循环部分。

根据图 2 加权切换系统与图 3 的 Büchi 自动机,可以得到图 4 所示的任务可行网络拓扑。图 2 中加权切换系统  $T$  有 4 个状态,图 3 的 Büchi 自动机  $A_\phi$  存在 9 个状态,所以通过笛卡尔乘积,Product 自动机  $A_P$  存在 36 个状态。在图 4 中,包含  $S_0$  的第一列是 Product 自动机的初始状态,包含  $S_8$  的最后一列为 Product 自动机的最终接收状态;图 4 中每一列代表同一 Büchi 自动机状态下的切换关系,且不同列的同一 Büchi 自动机状态下的切换关系是一致的,图中箭头代表两状态间的切换关系,箭头上的数字是两状态间的切换权重值,该值继承加权切换系统  $T$  的切换权重,为了图 4 的可读性,同一 Büchi 自动机状态下的切换关系仅在第一列做了图示,其余列与该列一致。

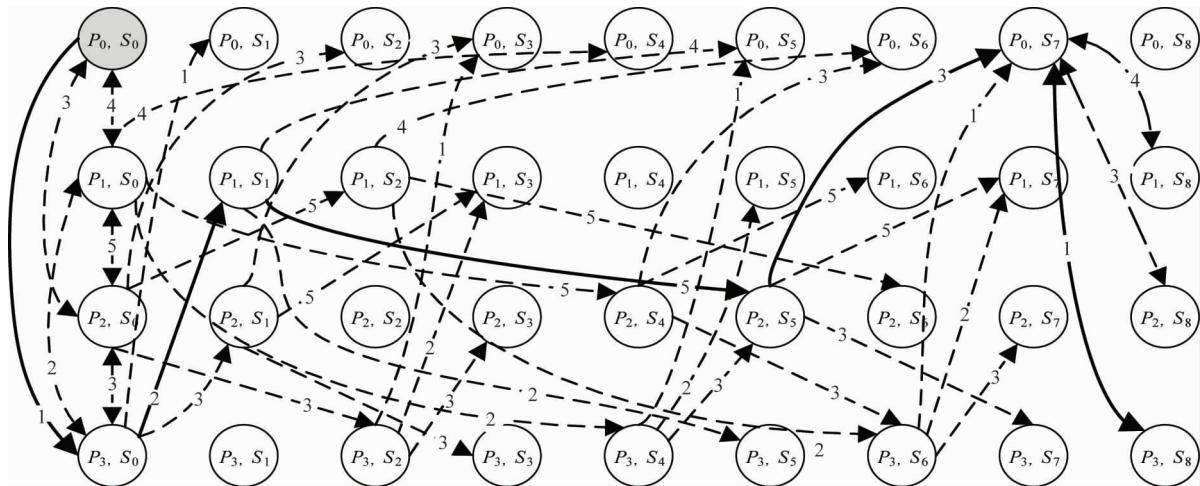


图 4 任务可行网络拓扑

在构建任务可行网络拓扑  $A_p$  后,根据搜索起始状态,最终接收状态和状态间的切换关系,利用 Dijkstra 最短路径搜索算法搜索最终的可行离散路径,算法过程如算法 2 所示。

算法 2: 搜索最优路径  $r_p$  (OptimalPath())

输入:  $T, A_\phi$ , 搜索起点  $s_{p0} = (q_0, s_0)$

输出: 可行最优路径  $r_p$

- (1) 构建任务可行网络拓扑  $A_p = T \otimes A_\phi, S_{p0} = s_{p0}$
- (2) 如果  $F_p = \emptyset$
- (3) 返回步骤(1)
- (4) 对于  $A_p$  每一个最终接收状态  $f_p \in F_p$
- (5) 搜索可行路径  $r_p = \text{Dijkstra}(s_{p0}, f_p)$
- (6) 如果  $r_p = \emptyset$
- (7) 返回路径不存在
- (8) 结束判断
- (9) 结束循环
- (10)  $r_p = \{r_p' \mid \min \text{Cost}(r_p'), r_p' \in r_p\}$
- (11) 返回最优路径  $r_p$

根据图 2 加权切换系统  $T$  与图 3 的 Büchi 自动机  $A_\phi$ , 设置搜索起点  $(P_0, S_0)$ , 利用算法 2 可以获得最优路径  $r_p$ , 如图 4 中黑色实线箭头所示,  $r_p = (P_0, S_0) \rightarrow (P_3, S_0) \rightarrow (P_1, S_1) \rightarrow (P_2, S_5) \rightarrow (P_0, S_7)$ , 图中  $S_7$  与  $S_8$  之间的切换源于任务公式(1)  $\text{GF}P_0$  的循环路径且  $\text{inf}(r_p) \cap F_p \neq \emptyset$ , 故该搜索路径  $r_p$  可被接受, 同时此路径的总耗费  $\text{Cost}(r_p) = 11$  是所有可行路径中最小的, 确保了路径的最优化。

**引理 1** Product 自动机路径映射<sup>[10]</sup>

对于在任务可行网络拓扑上搜索出的满足任务需求的任意路径  $r_p = (p_0, s_0) \rightarrow (p_1, s_1) \rightarrow (p_2, s_2) \dots$ , 在加权切换系统  $T$  中都有与之对应的路径  $r_T = p_0 \rightarrow p_1 \rightarrow p_2 \dots$  存在, 且  $r_T$  同样满足任务需求,  $r_T$  与  $r_p$  的总耗费相同。

根据引理 1, 图 4 中满足派送任务的最优路径  $r_p = (P_0, S_0) \rightarrow (P_3, S_0) \rightarrow (P_1, S_1) \rightarrow (P_2, S_5) \rightarrow (P_0, S_7)$  映射回加权切换系统中的最优路径为  $r_T = P_0 \rightarrow P_3 \rightarrow P_1 \rightarrow P_2 \rightarrow P_0$ , 该路径满足派送任务需求的同时保证了路径最优化。最后在 Android

平台上, 基于百度地图开发包的两点间的驾驶导航函数(即算法 2 中的  $BmapDriving()$ )将映射回加权切换系统中的离散路径  $r_T$  连续化, 如图 5 所示, 进行二次规划获得派送员实际可驾驶派送路线  $R$ , 二次规划的权重问题在算法一中已经被考虑。实现过程如算法 3 所示。

算法 3: 离散路径连续化(ProjectToR())

输入: 加权切换系统中的离散路径  $r_T$

输出: 快递员实际驾驶派送路线  $R$

- (1) 对于  $i = 0, 1, 2, \dots, n - 1, n$  为  $r_T$  路径节点数
- (2) 如果  $i = n - 1$
- (3)  $R(n - 1) = BmapDriving(r_T(n - 1), r_T(0))$
- (4) 否则  $R(i) = BmapDriving(r_T(i), r_T(i + 1))$
- (5) 判断结束
- (6) 结束循环
- (7) 返回  $R = R(0)R(1)R(2)\dots R(n - 1)$

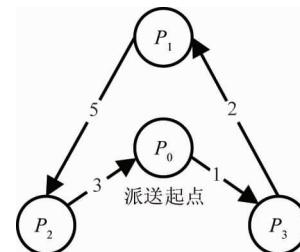


图 5 派送最优路径

### 3.4 快递派送最优路径规划方法

本文针对目前快递派送的区域受限与多点快递派送任务的派送路径很难到达最优问题, 考虑互联网+的智慧出行, 基于 Android 平台, 结合百度地图开发包的定位与导航, 提出了基于 LTL 的移动端快件派送路径规划方法, 提高快递员派送效率, 实现智慧派送。本方法首先根据快递员的派送任务, 利用算法 1 构建有限状态的加权切换系统 WFTS, 接着利用线性时序逻辑语言描述派送任务, 获得线性时序逻辑任务公式并通过 LTL2BA 工具包将任务公式转化为 Büchi 自动机  $A_\phi$ , 然后通过笛卡尔乘积构建任务可行网络拓扑  $A_p$  并在该拓扑上利用算法 2 搜

索最优离散路径  $r_p$ ,之后将  $r_p$  映射回加权切换系统,获得实际环境中的离散路径  $r_T$ ,最后基于百度地图开发包进行二次点对点路径规划,将离散路径转化为连续的快递员实际驾驶派送路线  $R$ 。

#### 算法 4: 快递派送最优路径搜索

输入: 派送起点  $P_0$ , 派送地点集  $P$ , 派送任务的线性时序逻辑公式  $\phi$

输出: 快递派送最优驾驶路径  $R$

- (1) 调用  $\text{Construct}T()$  算法构建加权切换系统  $T$
- (2) 构建 Büchi 自动机  $A_\phi = LTL2BA(\phi)$
- (3) 构建 Product 自动机  $A_p = T \otimes A_\phi$
- (4) 调用  $\text{OptimalPath}()$  算法搜索最优路径  $r_p$
- (5) 将  $r_p$  映射回加权切换系统得到  $r_T$
- (6) 调用  $\text{ProjectToR}()$  算法得到  $R$
- (7) 返回快递派送最优驾驶路径  $R$

## 4 路径规划实验

本小节结合具体实例,在 Android 平台上进行实验验证,如图 6 所示快递派送操作界面,操作界面中的一键获取快件信息按钮可以获得本次快递派送的任务地点,一键群发短信按钮通知取件人快递正在派送,获取最优路径按钮会调用算法 4 规划出快递员此次派单的派送路线。如图 7 所示,通过一键



图 6 快递派送操作界面



图 7 派送地图

获取快件信息按钮得到的派送地图,图中 A 为浙江工业大学屏峰校区、B 为浙江大学紫金港校区、C 为浙江大学玉泉校区、D 为浙江工业大学朝晖校区、E 为杭州汽车北站、F 为浙江省药品审评中心、G 为申花公园、H 为文新公园 8 个点为快递员当前的派送地址。快递员需要完成的派送任务为“派送员从 A 点出发,遍历 B、C、D、E、F、G 和 H 等 7 个地点进行派送快递,最后返回 A 点”。

$T.\text{adj} =$

inf	10	9.3	14.5	12.5	9.7	11.6	9.4
10	inf	5.7	7.7	2.9	3.8	2.2	2.0
9.3	5.7	inf	5.2	6.0	2.0	5.0	3.7
14.5	7.7	5.2	inf	5.8	5.2	5.7	6.5
12.5	2.9	6.0	5.8	inf	4.1	1.1	3.3
9.7	3.8	2.0	5.2	4.1	inf	3.1	1.8
11.6	2.2	5.0	5.7	1.1	3.1	inf	2.2
9.4	2.0	6.5	3.3	1.8	2.2	2.2	inf

首先根据快递员派送任务,利用环境建模算法  $\text{Construct}T()$  构建相应的加权切换系统  $T$ , 得到  $8 \times 8$  的邻接矩阵  $T.\text{adj}$ , 其中第一行代表点 A 依次到达 A、B、C、D、E、F、G 和 H 的实际距离,第二行代表点 B 依次到达 A、B、C、D、E、F、G 和 H 的实际距离,依此类推,距离单位为 km。

其次利用线性时序逻辑语言描述派送任务需

求。令搜索起点  $T, q_0 = p_0 = A, p_1, p_2, p_3, p_4, p_5, p_6$  和  $p_7$  分别代表 B、C、D、E、F、G 和 H 7 个地点, 则任务公式为

$$\phi = Fp_1 \wedge Fp_2 \wedge Fp_3 \wedge Fp_4 \wedge Fp_5 \wedge Fp_6 \wedge Fp_7 \wedge GFp_0 \quad (2)$$

接着将任务公式(2)利用 LTL2BA 工具包转换为 Büchi 自动机  $A_\phi$ , 并将加权切换系统  $T$  与 Büchi 自动机  $A_\phi$  作笛卡尔乘积得到任务可行网络拓扑  $A_P = T \otimes A_\phi$ , 然后通过算法二 OptimalPath() 搜索最优路径获得最优离散路径  $r_T$  并映射回加权切换系统得到  $r_T = p_0 p_2 p_5 p_3 p_4 p_6 p_1 p_7 p_0$ , 即  $r_T = A \rightarrow C \rightarrow F \rightarrow D \rightarrow E \rightarrow G \rightarrow B \rightarrow H \rightarrow A$  其总耗费  $Cost(r_T)$  为  $9.3 + 2 + 5.2 + 5.8 + 1.1 + 2.2 + 2 + 9.4 = 37\text{km}$ , 在所有可行路线中路程最短, 为最优路线。

最后通过 ProjectToR() 算法进行二次规划, 将离散的最优路径  $r_T$  连续化得到 Android 平台的快递员实际派送最优路径  $R$ , 结果如图 8 所示。图中蓝线为派送导航路线, 箭头为派送顺序。



图 8 实际派送最优路径

为进一步验证本方法的有效性, 与改进的  $A^*$  方法仿真结果进行比较<sup>[20]</sup>。 $A^*$  算法是对 Dijkstra 算法的改进,  $A^*$  算法加入启发信息进行搜索方向的引导, 避免一些无关节点搜索, 从而提高路径搜索效率, 属于启发式搜索所算法,  $A^*$  估价函数为  $f(n) =$

$g(n) + h(n)$ , 其中  $g(n)$  为从搜索结点到当前结点  $n$  的实际距离,  $h(n)$  为当前节点距离目标节点的估计值, 通常选取欧氏距离或者曼哈顿距离。但是  $A^*$  主要应用于点对点的路径规划, 无法直接应用于多点遍历的 TSP 问题, 为满足派单遍历性任务,  $A^*$  的  $h(n) = \min \times (node\_num - depth)$ , 其中  $depth$  为当前节点  $n$  的深度,  $node\_num$  为任务节点总数,  $(node\_num - depth)$  表示还需要遍历节点个数,  $\min$  表示节点  $n$  距离剩余节点的所有路径长度的最小值。实现步骤如下:

(1) 假设任务结点的个数为  $node\_num$ , 获取开始结点, 计算所有成员变量的权重, 将开始结点放入 open 表中;

(2) 如果 open 表不为空, 转步骤(3), 否则转步骤(7);

(3) 将 open 表中的估价  $f$  值最小的元素放入 close 表, 并将该元素从 open 表中删除;

(4) 获取当前节点的  $depth$ , 如果  $depth \geq node\_num + 1$ , 则找到最佳路径, 转步骤(7);

(5) 如果  $depth = node\_num + 1$ , 还访问一个结点到达目标结点, 设置初始结点的访问状态为 false, 表示初始结点没有被访问, 即可以回到出发点;

(6) 如果  $depth < node\_num + 1$ , 更新从当前结点出发可访问的结点和 open 表中剩余的结点的启发函数  $f$  值, 然后转步骤(2);

(7) 获取 close 表中的最后一个元素, 返回最佳路径与最短的距离值。

在图 7 中快递员需要完成的派送任务为“派送员从 A 点出发, 遍历 B、C、D、E、F、G 和 H 7 个地点进行派送快递, 最后返回 A 点”。根据此方法可以搜索出可行离散派送路径为  $r_T = A \rightarrow C \rightarrow F \rightarrow H \rightarrow B \rightarrow G \rightarrow D \rightarrow E \rightarrow A$ , 其总耗费  $Cost(r_T)$  为  $9.3 + 2 + 1.8 + 2 + 2.2 + 5.7 + 5.8 + 12.5 = 41.3\text{km}$ , 将该离散路径映射回百度地图得到快递员实际可行的连续派送路线如图 9 所示。通过对比可以发现, 改进  $A^*$  方法搜索所得总耗费为  $41.3\text{km}$  大于本文方法搜索的总耗费  $37\text{km}$ , 本文提出的方法能够获得最短的快递员派送路线。

图 9 改进 A<sup>\*</sup> 规划结果

本文提出的方法能够获得派送最优路径从而提高快递员的整体派送效率,但不得不承认当派送地点较多时,最优路径的搜索效率受到任务可行网络拓扑规模的影响,针对派送地点个数不同,实验计算得到了各最优路径的搜索时间,如表 1 所示,从表 1 可以看出当派送地点个数超过 7 后,路径搜索的效率很低,会影响快递员整体的派送效率。

表 1 搜索时间

派送地点数(个)	搜索时间(s)
2	0.002
3	0.005
4	0.011
5	0.048
6	0.246
7	1.024
8	4.855
9	30.623

## 5 结 论

本文在 Android 平台,利用百度地图开发包进行定位与导航,基于线性时序逻辑理论进行快递员多点派送任务的最优路径规划,根据派送任务,提出了相应的加权切换系统建模方法,避开百度地图实

际街道的繁琐建模,并在任务可行网络拓扑上采用 Dijkstra 最短路径搜索算法搜索最优路径,而非次优路径,保证了派送路径的最优性,而且在离散路径连续化时,由于环境建模采用百度地图自驾驶模型,最后生成的连续路径满足派送任务的同时符合快递员实际派送情形。该方法解决了目前快递派送的区域受限问题和派送路径非最优问题,提高了快递派送效率,实现了互联网+的智慧派送。

此外,该方法适用于派单任务点不是很多的情况,当任务点较多时,通过切换系统与 Büchi 自动机笛卡尔乘积构建的任务可行网络拓扑的状态数成倍增加,虽然能获得最优路径,但搜索效率受到很大影响,针对众多任务点的大型派送网络最优路径规划是今后的一个研究方向。

## 参考文献

- [1] 郑孝庭. 派送环节是赢得电子商务市场的关键. 中国物流与采购, 2010, 24: 56-57
- [2] 丁浩, 裴道方. 基于 Dijkstra 算法的快递车辆配送路径优化. 价值工程, 2014, 3: 15-18
- [3] 徐彬, 王权锋, 刘斌等. 贪婪和 A-Star 算法在物流配送中的应用及仿真. 软件, 2013, 34(6): 35-39
- [4] Emerson E A. Temporal and modal logic. Handbook of Theoretical Computer Science(Volume B): Formal Models and Semantics. Cambridge, MA: MIT Press, 1990. 995-1072
- [5] Vasile C I, Belta C. Sampling-based temporal logic path planning. In: Proceedings of the Intelligent Robots and Systems, Tokyo, Japan, 2013. 4817-4822
- [6] Fainekos G E, Kress-Gazit H, Pappas G J. Hybrid controllers for path planning: a temporal logic approach. In: Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 2005. 4885-4890
- [7] Guo M, Johansson K H, Dimarogonas D V. Motion and action planning under LTL specifications using navigation functions and action description language. In: Proceedings of the Intelligent Robots and Systems, Tokyo, Japan, 2013. 240-245
- [8] Fainekos G E, Kress-Gazit H, Pappas G J. Temporal logic motion planning for mobile robots. In: Proceedings of IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005. 2020-2025
- [9] Cimatti A, Clarke E, Giunchiglia E, et al. Nusmv 2: an opensource tool for symbolic model checking. In: Proceedings of the International Conference on Computer Ai-

- ded Verification, Copenhagen, Denmark, 2002. 359-364
- [10] Smith S L, Tumova J, Belta C, et al. Optimal path planning under temporal logic constraints. In: Proceedings of the Intelligent Robots and Systems, Taipei, China, 2010. 3288-3293
- [11] Smith S L, Tumova J, Belta C, et al. Optimal path planning for surveillance with temporal-logic constraints. *International Journal of Robotics Research*, 2011, 30(14): 1695-1708
- [12] 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
- [13] 禹鑫焱, 陈浩, 郭永奎等. 基于线性时序逻辑理论的仓储机器人路径规划. 高技术通讯, 2016, 26(01): 16-23
- [14] 丁洁. 基于深度优先搜索算法的快递派送策略研究. 市场周刊(理论研究), 2012, 1: 21-23
- [15] Wang K P, Huang L, Zhou C G, et al. Particle swarm optimization for traveling salesman problem. In: Proceed-
- ings of International Conference on Machine Learning and Cybernetics, Xi'an, China, 2003. 1583-1585
- [16] 鲍立婷. 粒子群算法在基于 LBS 快递派送中的应用研究: [硕士学位论文]. 南昌:东华理工大学信息工程学院, 2016
- [17] 杨浩雄, 刘彤. 商业区快递车辆路径优化及仿真. 计算机仿真, 2014, 31(08): 443-451
- [18] 覃运梅, 毛海军, 黑秀玲. 基于自动快递机的快递配送车辆路径优化研究. 公路交通科技, 2015, 32(10): 134-140
- [19] Gastin P, Oddoux D. Fast LTL to Büchi automata translation. In: Proceedings of the 13th International Conference on Computer Aided Verification, Paris, France, 2001. 53-65
- [20] 孟超. 具备自动优化拣货路径功能的仓库管理系统的  
设计与实现: [硕士学位论文]. 哈尔滨: 哈尔滨工业  
大学软件学院, 2013

## Express delivery path planning for mobile terminals based on linear temporal logic

Yu Xinyi, Guo Yongkui, Ou Linlin, Wang Tao, Lu Liang, Zhang Aimei

( College of Information Engineering, Zhejiang University of Technology, Hangzhou 310000 )

### Abstract

The path planning of express delivery was studied. Based on linear temporal logic (LTL), a multi-point express delivery path planning method for mobile terminals was proposed to solve the problems of large number of delivery points in the delivery area and the courier's low familiarity with the delivery area. The method solves the problems of location and navigation by using the development package of the Baidu map on mobile phones. The proposed method can be divided into top path planning and bottom path planning. The top path planning translates the express delivery problem into the traveling salesman problem (TSP). For avoiding the difficult and huge modeling of the actual road environment in the Baidu map, a weighted finite-state transition system is constructed by the express delivery locations. The transition weights between the states are on the basis of driving distance of the Baidu map rather than the straight-line distance, so that the bottom path planning results meet the actual situation and its optimality is guaranteed. The multi-point express delivery task is described by a linear temporal logic formula. Then, a feasible network topology related to the task is built by combining the above transition system with a linear temporal logic formula. The discrete optimal path of the courier can be obtained by using the Dijkstra algorithm in the network topology. For the bottom path planning, the actual continuous path between any two adjacent path points of the discrete path is generated based on the Baidu map. Thus, the discrete path in top path planning is translated to continuous trajectory to satisfy the courier express delivery task. The experimental results show that the proposed method can solve the delivery path planning problem of the large number of delivery points and the familiarity restriction of the courier for the delivery area, which ensures delivery route optimality.

**Key words:** express delivery, path planning, linear temporal logic (LTC), mobile terminal