



基于拥堵预测的路径规划算法研究

杨海芬*, 张 辉, 蒋贤皓, 林水生

(电子科技大学信息与通信工程学院, 成都 611731)

摘要 随着城市出行车辆数的急剧增加, 城市交通路网拥堵日益严重, 因此研究路网通行效率最大化、个人通行时间和路径最优化具有重要意义。目前智慧交通大多采用周期性监测拥堵并重新规划路径的方法, 这类方法得到的最优路径往往导致下次重规划的绕行, 不利于全局的最优化。本文提出一种基于拥堵预测的路径规划算法, 通过分析大量历史交通流数据的时空相关性预测未来时刻的交通流参数, 再通过熵权法和模糊综合判定得出拥堵状态分级, 之后综合考虑当前时刻和未来时刻拥堵状况从而规划出更加合理的路径。仿真结果表明, 与基于重规划的动态路径规划算法相比, 本算法在行驶时间上减少 5.35%, 在行驶距离上减少 11.72%。

关键词 智慧交通; 拥堵预测; 路径规划

中图分类号 TN919

文献标志码 A

DOI 10.12178/1001-0548.2022333

Research on route planning based on congestion prediction

Yang Haifen*, Zhang Hui, Jiang Xianhao, and Lin Shuisheng

(School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract With the rapid increase of vehicles, the road network has become increasingly congested. It is of great significance to study the maximization of road network traffic efficiency, personal traffic time and path optimization. The existing methods periodically monitor congestion and replan paths. The previous path obtained by this kind of algorithm often leads to the detour of the next replanning, which is not conducive to the global optimization. This paper proposes a path planning algorithm based on congestion prediction. By analyzing the temporal and spatial correlation in a large number of historical traffic flow data, the traffic flow parameters at the future time are predicted, and then the congestion state classification is obtained through entropy weight method and fuzzy comprehensive judgment. Then a more reasonable path is planned by comprehensively considering the current and future congestion conditions. The simulation results show that compared with the dynamic path planning algorithm, this algorithm reduces the travel time by 5.35% and the travel distance by 11.72%.

Key words smart transportation; congestion prediction; route planning

随着社会的飞速发展, 我国私家车的数量也逐年增加。同时, 汽车登记数量和机动车驾驶人数量的上升, 使交通拥堵现象愈发明显。智慧交通系统旨在借助先进的信息与通信技术提供高效快捷的出行服务, 其中, 路径规划作为智慧交通系统的核心技术之一, 它的研究进展直接影响着智慧交通系统的部署。

目前, 路径规划算法主要分为静态路径规划和动态路径规划。在静态路径规划中, 路段的权值不会随着时间发生变化, 典型的算法如 Dijkstra 算法^[1]、A*算法^[2]、蚁群算法^[3]等等。Xu 等使用双向搜索

策略来改进传统 A*算法来加快搜索的效率^[4]。Lee 等人利用遗传算法改进蚁群算法中的影响信息素的参数, 提高了算法的收敛性能和最优解寻找能力^[5]。但是路网具有动态性, 前一时刻的还畅通无阻的路段, 有可能在车辆到达时变得拥堵, 而固定的路段权值很难适应动态的路网, 难以避开拥堵。相对于静态路径规划, 动态路径规划算法考虑了时变的路段权值, 在行进过程中不断获取实时路况, 重新计算路段的权值并更新修正路线, 以此避开拥堵, 如典型 D*算法。Ramkumar 研究了基于车载自组织网络 VANET 的动态路径规划, 降低了车辆的行驶

收稿日期: ; 修回日期:

基金项目: 国家重点研发计划 (2018YFB2101303)

作者简介: (1977—), 女, 博士, 副教授, 主要从事智能算法、通信与信号方面的研究。

成本和提高了避开拥堵的能力^[6], Noguchi 利用人工势场法结合强化学习的方式在动态环境中得到安全的路径^[7]。虽然动态路径规划算法通过周期性重规划的措施可以在一定程度上避开拥堵,但只考虑当前时刻的路况,没有从全局角度出发考虑路径在未来时刻的潜力,很难得到全局最优的路径^[8]。

本文通过分析大量历史交通流数据的时空相关性预测未来时刻的交通流参数,再通过熵权法和模糊综合判定得出拥堵状态分级,之后综合考虑当前时刻和未来时刻拥堵状况规划出更加合理的路径。

1 相关理论基础和模型

交通流预测通过分析历史交通流中的流动趋势,对未来时刻的交通流给出预测结果,可以看做是一类序列预测问题。然而,交通流序列呈现出一种复杂和动态的时空相关性^[1],对于路网中的每一个节点,在同一时刻,邻居节点会对它影响,这是空间相关性。此外,每一个节点也可能受到过去时刻自己的影响,这是时间相关性。

在空间相关性建模方面,目前的主流方案是 ChebNet 图卷积^[2],卷积核 Θ 和图信号 x 在时域上的卷积结果如式 (1) 所示。

$$\Theta *_{\text{g}} x = \sum_{k=0}^{K-1} \theta_k T_k(\bar{L}) x \quad (1)$$

其中 θ_k 是切比雪夫多项式系数, K 是卷积核的感受野半径, $L = D - A$ 是图的拉普拉斯矩阵,由邻接矩阵 A 和度矩阵 D 计算得来。 $\bar{L} = 2L/\lambda_{\max} - I_n$ 是归一化的拉普拉斯矩阵, λ_{\max} 是拉普拉斯矩阵的最大特征值。 $T_k(\bar{L})$ 是第 k 阶的切比雪夫多项式,其中 $T_0(\bar{L}) = I_n$, $T_1(\bar{L}) = \bar{L}$,之后的结果通过式 (2) 进行推导。

$$T_k(\bar{L}) = 2\bar{L}T_{k-1}(\bar{L}) - T_{k-2}(\bar{L}) \quad (2)$$

式 (2) 中,由于 $T_0(\bar{L}) = I_n$, $T_0(\bar{L})x$ 使得图中每个节点都提取了自己的信息。 $T_1(\bar{L})$ 中的所有非零元素都是 1 阶邻居和当前节点本身,如果将本身作为 0 阶邻居节点,那么 $T_1(\bar{L})x$ 使得图中每个节点都提取了 0 阶到 1 阶邻居的信息。以此类推, $T_k(\bar{L})x$ 使得每个节点都提取了 0 阶到 k 阶邻居的信息。最终这些信息通过切比雪夫多项式系数 θ_k 进行加权求和,这些系数决定哪一阶的邻居提供更多的信息。而同一阶邻居对当前节点的不同影响程度由邻接矩阵 A 进行区分。

在时间相关性建模方面,目前主流的方案主要

有循环卷积神经网络 RNN^[11] 和卷积神经网络 CNN^[4]。CNN 模型一般空间维度的卷积核大小为 1,步长为 1,使得只在时间维度聚集信息。时间维度的卷积核大小为 K_t ,步长为 1,使得每一个节点都聚集了当前时刻以及 $K_t - 1$ 个邻近历史时刻的信息。

路阻函数表述了路段的通行时间与路段上采集到的交通流参数之间的关系, BPR 路阻函数如式 (3) 所示^[13]:

$$t = t^0 \frac{2}{1 \pm \sqrt{1 - Q/C}} \quad (3)$$

其中, t 是通行时间, t^0 是自由行驶时的通行时间。 C 是路段通行能力,是流量 Q 能够达到的理论最大值,如式 (4) 所示:

$$C = Q_{\max} = \frac{V_f K_j}{4} \quad (4)$$

其中 K_j 是阻塞时的交通密度, V_f 是自由行驶时的通行速度。

2 拥堵预测模型设计

本文的拥堵预测模型主要包括交通流预测和拥堵状态判定两个阶段。交通流预测模型对路网中的时空相关性进行建模,然后对未来时刻的交通流参数进行预测。拥堵状态判定模块在给定交通流参数的情况下,对拥堵状态进行分级判定。

2.1 交通流预测模型设计

在 ChebNet 模型中,同一阶邻居节点对当前节点的不同影响程度由邻接矩阵区分,邻接矩阵的构建,主要有基于连接和基于距离这两种方式。基于连接的邻接矩阵只表达了连接关系,没有对同一阶的邻居节点做区分。基于距离的邻接矩阵单一使用距离作为区分标准,不够全面。

针对上述问题,本文采用三维权重矩阵 $W \in R^{K \times N \times N}$ 进行优化,其中第一个维度是切比雪夫多项式的阶数,后面两个维度为路网节点数量,并按照式 (5) 进行初始化。

$$w_{ij}^k = \begin{cases} 1 & T_k(\bar{L})_{ij} \neq 0 \\ 0 & \text{否则} \end{cases} \quad (5)$$

使用权重矩阵后的图卷积式如 (6) 所示。

$$\Theta *_{\text{g}} x = \sum_{k=0}^{K-1} \theta_k (w^k \odot T_k(\bar{L})) x \quad (6)$$

通过可训练参数 w^k , 可以自适应的调整同一阶邻居的影响程度。

时间相关性提取方面,本文改进了传统的

CNN 模型, 如图 1 所示。其中横轴代表空间信息, 表示当前路网中所有节点在同一时刻采集到的交通参数。纵轴代表时间信息, 从上往下是路网中每个节点严格按照时间顺序采集到的交通参数序列。在空间维度, 卷积核的长度为 1, 相当于在卷积的过程中只聚集时间维度的信息, 并且空间维度的卷积步长也为 1, 从而对路网中的所有节点都提取时间相关性。在时间维度, 使用不同大小的卷积核进行卷积。使用多个卷积核进行卷积后, 时间轴上越靠后的时刻能聚集到更多的信息, 从而尽可能的提高信息利用率, 增加卷积的感受野, 提高时间相关性的提取能力。

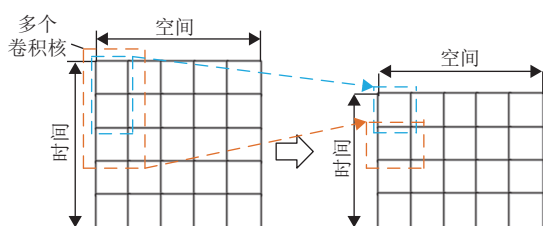


图 1 提取时间相关性的 CNN 模型

本文采取的交通流预测模型如图 2 所示, 使用时空图卷积层提取时空相关性。时空图卷积层由上述的一个图卷积和两个时间维度卷积堆叠而成。模型最后的输出通过时间维度卷积和全连接层转为为未来时刻的交通流参数。

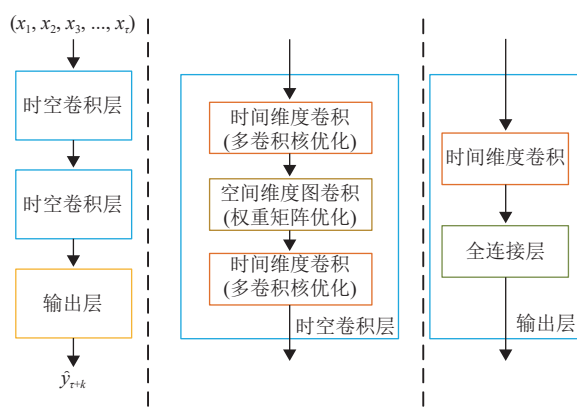


图 2 交通流预测模型

2.2 拥堵状态分级

拥堵状态评估主要分为单参数评估和多参数评估。单参数评估只使用到了一种交通流参数, 根据输入参数的取值得到所在的区间, 即为最终的拥堵状态判定结果。由于路网的复杂性, 单参数不能够完全反映路网的通行状况, 所以本文选取了多参数判定方案, 通过熵权法为影响因子更高的参数赋予

更高的权重, 最后利用模糊综合法判定, 综合考虑多参数后确定最终分级。具体步骤如下:

(a) 假设样本长度为 k , 每个样本有 n 种交通流参数, d_{il} 代表第 i 个样本的第 l 种交通流参数。

(b) 将样本归一化并计算第 i 个样本的第 l 种参数占该种参数的比重 p_{il} , 通过比重 p_{il} 计算出第 l 种参数的熵值 e_l 和差异系数 $g_l = 1 - e_l$, 如式 (7) 所示。

$$e_l = -\frac{1}{\ln(k) \sum_{i=1}^k p_{il} \ln(p_{il})} \quad (7)$$

(c) 通过差异系数确定第 l 种参数的权值, 某种交通流参数的差异系数越大, 说明参数对拥堵状态的影响更大, 如式 (8) 所示。

$$w_l = \frac{g_l}{\sum_{l=1}^n g_l} \quad (8)$$

确定好权重后, 通过模糊综合判定进行拥堵分级。模糊综合判定的核心思想是不绝对的肯定与否定, 而是利用隶属度来描述一个元素的归属问题, 隶属度越接近 1, 那认为该元素归的归属度更高。具体步骤如下:

(a) 确定每一种交通参数分级区间以及拥堵状态分级。假设有 m 种拥堵分级, $H = \{\mu_1, \dots, \mu_{2m-2}\}$ 是对应的临界值。

(b) 计算参数 d_{il} 对应每一种拥堵分级的隶属度 r_{il}^m , 采用梯形隶属度函数下的计算方式如图 3 所示。

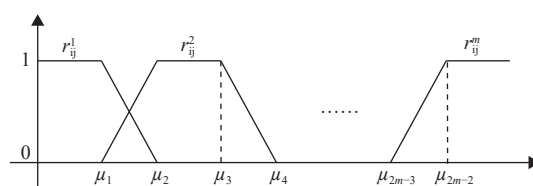


图 3 隶属度函数

(c) 综合考虑每一种参数, 得到第 i 个样本的评价矩阵 P_i , 如式 (9) 所示。

$$P_i = [w_1, \dots, w_n] \begin{bmatrix} r_{i1}^1 & \dots & r_{i1}^m \\ \dots & \dots & \dots \\ r_{in}^1 & \dots & r_{in}^m \end{bmatrix} = [p_1, \dots, p_m] \quad (9)$$

最终的拥堵分级为 $\max(p_1, \dots, p_m)$ 所在的下标。

3 基于拥堵预测的路径规划算法

在路径规划中, 计算路段的通行时间是非常重要的环节, 是影响车辆在路网中行驶的重要因素。在基于拥堵预测的场景下, 拥堵预测模型能够提供

未来时刻的交通流参数以及拥堵状态，通过路阻函数可以将路段上采集到的交通流量转化为路段的通行时间。路段通行时间如图4所示。

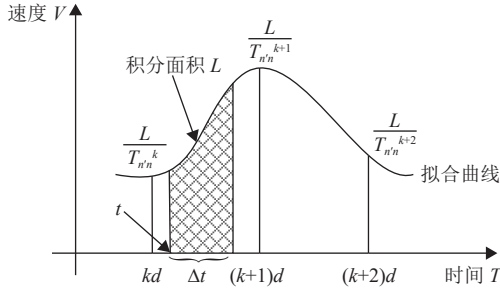


图4 路段通行时间

其中 d 是预测的时间间隔， L 是该路段的长度， t 是进入路段的时刻。通过拥堵预测模型和分段路阻函数能够得到未来每个时刻的瞬时通行时间 T_{ij}^k 和平均通行速度 L/T_{ij}^k ，之后利用拟合补全拥堵预测未提供的细节，采用积分的方式将离散的信息组合成连续的形式。再从 t 时刻开始积分，则路段在未来时刻的通过时间为积分面积 L 的多边形在时间轴上的投影。

上述路段通行时间的计算方法可以量化拥堵时长带来的影响，从而发现更有潜力的路段。并且能够及时规避拥堵，比如某些路段在进入时刻还是畅通状态，但在驶离阶段发生拥堵，那么这些影响会表现在路段的通行时间上，使得最短路径算法不再选择该路段，达到避开拥堵的目的。

计算完路段的通行时间后，路网中的每条路段都有了权重，之后可以通过路径规划算法搜索路径。

算法1 拥堵预测下的路径规划算法

- 1: 输入起点 n_s , 终点 n_g , 开始时刻 t
- 2: 通行时间矩阵 $T = \text{cal_travelT}(t)$
- 3: 初始化 $Q = \emptyset, rhs(n_s) = 0$
- 4: for $n \in N : g(n) = \infty, rhs(n) = \infty$
- 5: $Q.add(n_s, key(n_s))$
- 6: while $Q.minKey < key(n_g) \parallel rhs(n_g) \neq g(n_g)$
- 7: $n = Q.minNode()$
- 8: if $g(n) > rhs(n)$:
- 9: $g(n) = rhs(n)$
- 10: for $q \in Out(n) : \text{updateNode}(q, g(q), T)$
- 11: else:
- 12: $g(n) = \infty$
- 13: for $q \in Out(n) \cup \{n\} : \text{updateNode}(q, g(q), T)$

- 14: function $\text{updateNode}(n, g(n), T)$
- 15: if $n \neq n_g$:
- 16: $rhs(n) = \min_{n' \in In(n)} (g(n') + \text{cost}(n', n, g(n)))$
- 17: if $n \in Q : Q.remove(n)$
- 18: if $g(n) \neq rhs(n) : Q.add(n, key(n))$
- 19: end function

如算法1所示， $out(n)$ 表示经过节点 n 一跳能到达的节点， $in(n)$ 表示一跳能到达 n 的节点， $g(n)$ 表示上一轮迭代中到达节点 n 的最短时间。 $rhs(n)$ 表示当前迭代中到达节点 n 的最短时间，计算方式如第16行所示，其中 $\text{cost}(n', n, g(n))$ 是路段 (n', n) 通行时间。如果 $g(n) \neq rhs(n)$ 表示当前迭代中 $g(n)$ 未同步到最优；如果 $g(n) = rhs(n)$ ，节点 n 已经同步到最优。除此之外，启发因子 key 如式(10)所示。

$$key(n) = \min(g(n), rhs(n)) + h(n, n_g) \quad (10)$$

式中 $h(n, n_g)$ 是当前节点到终点的预估代价。

在算法的初始化阶段，通过拥堵预测模型和分段路阻函数计算路网的瞬时通行时间矩阵，并设置起点的 $rhs(n_s)$ 为0，其余节点的 $g(n)$ 和 $rhs(n)$ 都为 ∞ ，并将起点加入优先队列，如第1到5行所示。

之后从起点开始搜索迭代。每次迭代从优先队列取出 key 值最小的节点，如果 $g(n) > rhs(n)$ ，说明上游节点找到了更优的路径，此时需要同步节点 n ，并采用广度优先搜索策略，并将影响向下游节点传播。如果 $g(n) < rhs(n)$ ，则当前节点信息有误，更新当前节点和其下游节点。

搜索迭代过程一致持续到发现终点完成同步并且其 key 是优先队列中的最小值。当迭代结束后，通过判断 $g(n_g)$ 是否为 ∞ 可得是否找到路径，并且从终点不断回溯上游节点 n' 中 $g(n') + \text{cost}(n', n)$ 最小的节点即可得到最优路径。

4 仿真结果及分析

本文使用的地图数据来源于开放街道图OpenStreetMap，下载江苏省无锡市巡塘镇的部分地图，对原始地图进行清洗。之后使用交通仿真软件SUMO产生交通流并添加检测点，获取仿真期间路网上的交通流量、通行速度、交通密度等交通流参数。在交通流预测阶段，使用平均绝对误差MAE、平均绝对百分比误差MAPE、均方根误差RMSE作为准确度指标。在拥堵预测阶段，将拥堵状态划分为畅通、基本畅通、轻度拥堵、重度拥堵4个分级。仿真参数设置如表1所示，其中地图

面积为在 OSM 官网中选取地图所对应的真实地图面积, 选取路网中的主干路及其支路作为检查点。每轮实验进行 10 轮迭代, 每次迭代投入 20 辆测试车辆, 在所有的测试车辆到达终点后, 统计各项指标的平均值。

表 1 仿真参数

参数	取值
切比雪夫阶数	3
时间卷积最小卷积核大小	3
历史数据长度	1h
地图面积	8.2平方公里
路网道路数量	2 224条
检测点数量	552个

表 2~4 给出了不同预测模型下交通流量、交通密度、平均通行速度的预测结果。可以看出, LSTM 模型预测效果好于传统 SVR 模型, 而本模型和 STGCN 模型则优于 LSTM 和 SVR 模型。这是因为 LSTM 和 SVR 只提取时间相关性, 而本模型和 STGCN 还考虑了路网拓扑方面的空间相关性。

表 2 交通流量预测结果 (单位: vec/h)

模型	MAE	MAPE(%)	RMSE
本模型	23.68	9.19	31.36
STGCN	24.85	9.62	32.52
LSTM	28.35	12.55	35.46
SVR	29.48	12.69	37.76

表 3 交通密度预测结果 (单位: %)

模型	MAE	MAPE(%)	RMSE
本模型	1.23	9.21	0.022 2
STGCN	1.26	9.7	0.025 3
LSTM	1.32	11.5	0.025 4
SVR	1.37	11.16	0.026 3

表 4 平均通行速度 (单位: km/h)

模型	MAE	MAPE(%)	RMSE
本模型	1.59	2.96	2.57
STGCN	1.64	3.09	2.68
LSTM	1.97	3.81	3.79
SVR	2.01	3.91	3.46

图 5 给出了本文模型在高峰时期的拥堵预测细节图。可以看到拥堵预测的结果和实际状况较为吻合, 包括 9 点至 10 点半期间畅通到重度拥堵的转变, 以及 1 点至 1 点半期间重度拥堵到畅通的过

渡, 都能被本模型大致捕捉到。在 8 点至 2 点 15 的 75 个采样点中, 畅通状态的预测错误次数为 5 次, 基本畅通状态的预测错误次数为 5 次, 轻度拥堵状态的预测错误次数为 4 次, 总计高峰时期的预测准确度为 81.33%。

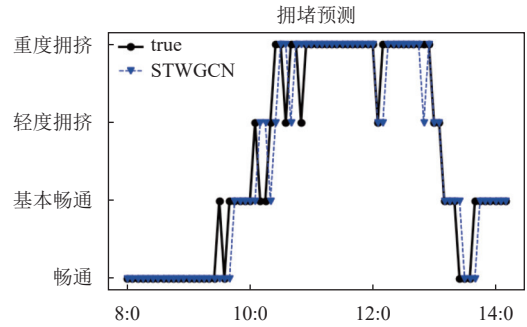


图 5 拥堵预测细节

图 6 给出了不同数量车辆情况下不同算法的行驶时间对比。可以发现, 随着投入车辆的增加, Dijkstra 算法的劣势逐渐显现, 因为在当前时刻的最优路径并不一定是未来时刻的合理路径, 导致后续的行驶过程中遇上拥堵从而影响了行驶时间。D*算法在每个交叉路口重新规划路径, 从而避开了拥堵, 提高了行驶效率。而本文的算法因为结合了当前时刻和未来时刻的路况, 其长期收益更大。在投入 900 辆车时, 本文算法相对于 Dijkstra 算法在行驶时间上减少了 22.13%, 相对于 D*算法减少了 5.35%。

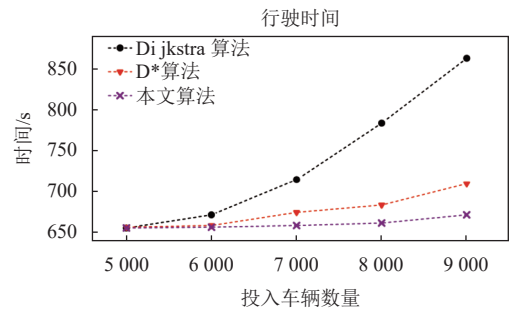


图 6 行驶时间

表 5 给出了预期行驶时间与实际行驶时间对比。可以看到 Dijkstra 算法采用当前时刻路况计算路段通行时间, 因此预期行驶时间与实际行驶时间的差值会随着路况的复杂变化而急剧增大。D*算法会不断重规划进行修正, 所以差值相对 Dijkstra 算法较小但波动性强。本文算法考虑了未来时刻路况并且完整的利用了路况的变化趋势, 因此差值能稳定的维持在一个较小的范围。

表 5 实际与预期行驶时间对比 (单位: 时间 (s))

车辆数		5 000	6 000	7 000	8 000	9 000
Dijkstra	实际	656	672	715	784	863
	预期	588	591	583	596	605
D*	实际	657	659	675	684	710
	预期	586	588	647	605	657
本文	实际	656	657	659	662	672
	预期	621	629	621	630	645

图 7 给出了不同算法的行驶距离对比图。可以明显看到, D*算法行驶距离最长, 这是因为频繁的重新规划带来了绕行。Dijkstra 算法的行驶距离最短, 但是最容易遇上拥堵。本文的算法能在行驶时间比 D*少的情况下也有更短的行驶距离, 在投入数量为 9 000 辆车时, 行驶距离比 D*算法减少了 11.72%。

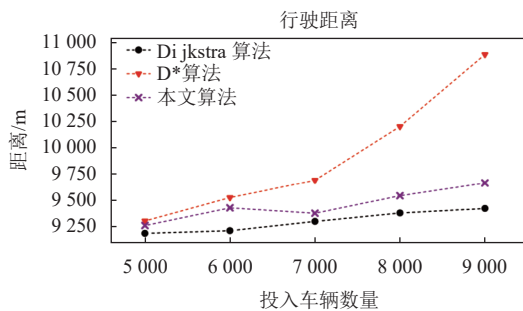


图 7 行驶距离

5 结束语

本文利用权重矩阵和多卷积核提高交通流预测的准确度, 并采用熵权法和模糊综合判定进行拥堵分级输出, 在此基础上计算未来时刻路段的通行时间并进行路径规划。仿真表明该种方式能更准确的计算路段通行时间, 从而降低路径规划算法的实际行驶时间和行驶距离, 为智慧交通提供一种可行的路径规划方案。

参考文献

- [1] Zhu Z, Li L, Wu W, et al. Application of improved Dijkstra algorithm in intelligent ship path planning[C]. Chinese Control and Decision Conference, Kunming, 2021: 4926-4931
- [2] Candra A, Budiman M A, Hartanto K. Dijkstra's and AStar in finding the shortest path: a tutorial[C]. International Conference on Data Science, Artificial Intelligence and Business Analytics, Medan, 2020: 28-32
- [3] Jangra R, Kait R. Analysis and comparison among ant system ant colony system and Max-Min ant system with different parameters setting[C]. International Conference on Computational Intelligence & Communication Technology, Ghaziabad, 2017: 1-4
- [4] Xu Z, Liu X, Chen Q. Application of improved Astar algorithm in global path planning of unmanned vehicles[C]. Chinese Automation Congress, Hangzhou, 2019: 2075-2080
- [5] Lee M, Yu K. Dynamic path planning based on an improved ant colony optimization with genetic algorithm [C]. IEEE Asia-Pacific Conference on Antennas and Propagation, Auckland, 2018: 1-2
- [6] Ramkumar P, Uma R, Usha S, et al. Real time path planning using intelligent transportation system for hybrid VANET[C]. International Conference on Power Energy Control and Transmission Systems, Chennai, 2020: 1-7
- [7] Noguchi y, Maki T. Path planning method based on artificial potential field and reinforcement learning for intervention AUVs[C]. IEEE Underwater Technology, Kaohsiung, 2019: 1-6
- [8] Li J. A Traffic prediction enabled double rewarded value iteration network for route planning[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 4170-4181
- [9] Shi X, Chen Z, Wang H. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[C]. Annual Conference on Neural Information Processing Systems, Montreal, 2015: 802-810
- [10] Hammond D K, Vandergheynst P, Gribonval R. Wavelets on graphs via spectral graph theory[J]. Applied and Computational Harmonic Analysis, 2011, 30(2): 129-150
- [11] Madan R, Mangipudi P S. Predicting computer network traffic: a time series forecasting approach using DWT ARIMA and RNN[C]. Eleventh International Conference on Contemporary Computing, Noida, 2018: 1-5
- [12] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting[C]. International Joint Conference on Artificial Intelligence. Stockholm, 2018: 3634-3640
- [13] 周继彪, 王露, 孟现勇. 道路路阻函数模型及适用性研究 [J]. 交通信息与安全, 2013, 31(2): 14-17

编辑 刘飞阳