Journal of Computer Applications ISSN 1001-9081 2015-【-10
计算机应用,2015,【(【): 【-【 CODEN JYIIDU http://www.joca.cn

文章编号: 1001-9081(2015)【-0000-00 doi:10.11772/j.issn.1001-9081.2015.【.0000

一种具有交通规则约束的改进 Dijkstra 算法

任鹏飞[1]，秦贵和[1]，董劲男[1,2*]，郑啸天[1]，李滨[1]

（1. 吉林大学 计算机科学与技术学院，长春 130012；2. 吉林大学 通信工程学院，长春 130012）

(*通信作者电子邮箱 dongjinnan@jlu.edu.cn)

摘　要: 传统 Dijkstra 算法在路径规划时无法适用于具有交通规则约束的交通网络。为解决该问题，本文在以往的路网模型和算法的基础上，提出一种具有交通规则约束的改进 Dijkstra 算法。算法对节点新增“待选择状态”和“可再更新状态”，用以解决节点具有交通规则约束的问题；同时引入祖父节点，从而生成交通网络中各节点的三元组信息，以此作为回溯依据，可以得到从初始节点到目的节点的最短路径。该算法不仅适用于具有交通规则约束的交通网络，且具有较低的复杂度。通过理论分析证明了算法的正确性，并以长春市朝阳区的实际交通网络和随机添加的交通规则约束为数据进行了实验测试，验证了算法的有效性。

关键词: 智能交通；路径规划；Dijkstra 算法；交通规则；节点三元组

中图分类号: TP301.6 **文献标志码: A**

An Improved Dijkstra Algorithm with Traffic Rules Constraints

REN Pengfei[1],QIN Guihe[1],DONG Jinnan[1,2],ZHENG Xiaotian[1],LI Bin[1]

(1.College of Computer Science and Technology, Jilin University, Changchun 130012, China;

2. College of Communication Engineering, Jilin University, Changchun 130012, China)

Abstract: Traditional Dijkstra algorithm cannot adapt for the transportation network with traffic rules constraints during path planning. In order to solve this problem, based on the previous network models and algorithms, an improved Dijkstra algorithm with traffic rules constraints was proposed. It added new "to be selected state" and "renewable state" on the nodes, to solve the problem of nodes with traffic rules constraints. Meanwhile, grandfather node was introduced, thereby generating triples information of each node in the traffic network. Therefore, taking this as a backtrack basis, the shortest path from the starting node to the destination node could be obtained. This algorithm is not only applicable to the transportation network with traffic rules constraints but with low complexity. The correctness of the algorithm is verified through theoretical analysis. The effectiveness of the algorithm is verified through experimental tests carried out with actual traffic network in Chaoyang District of Changchun city and random added traffic rules constraints.

Keywords: intelligent transportation; path planning; dijkstra algorithm; traffic rules; triples of adjacent node

0 引言

近年来，随着地球信息科学的高速发展，智能交通系统(Intelligent Transportation System，ITS)逐渐成为了交通系统的发展方向，路径规划模块是智能交通系统的核心模块之一，对系统的可用性有着决定性的作用[1]。

众所周知，最短路径问题是网络优化中最广泛的问题之一，其核心就是算法的设计[2]。传统的 Dijkstra 算法是业界公认的求解最短路径问题的经典算法[3]，大部分最短路径规划算法都是在其基础上设计提出的[4-5]。然而在其应用于实际的交通网络中时，会忽略许多方面的重要信息，比如线路限制、交通规则等信息[6]，从而丢失交通网络的基本特征，返回错误的路径，因此必须对 Dijkstra 算法加以改进以寻得最短路径。

收稿日期:2015-05-05; **修回日期**:2015-06-02。

基金项目:2011 年物联网发展专项“支持车联网的车载信息系统及应用示范”(无编号)。

作者简介:任鹏飞(1990－)，男，内蒙古赤峰人，硕士研究生，研究方向:智能控制、嵌入式系统；秦贵和(1962－)，男，山东高密人，教授，博士，CCF 会员(E200044088M)，研究方向:智能控制、嵌入式系统、汽车电子与信息技术；董劲男(1980－)，男，吉林长春人，讲师，博士，研究方向:智能控制、嵌入式系统、网络控制；郑啸天（1990－），男，硕士研究生，陕西汉中人，研究方向:智能控制、嵌入式系统；李滨(1991－)，男，山东临沂人，硕士研究生，研究方向:智能控制、汽车电子与信息技术。

近年来国内外学者主要致力于研究对路网模型的改进和对算法改进的两种方向。其中，基于模型的方法主要有虚拟节点法、对偶法、弧段法等[7]。采用改进型 Dijkstra 算法的方法主要是在规划过程中考虑实际交通约束的影响[8]。两种改进方法在时间复杂度和空间复杂度上各有取舍。模型改进增大了地图的存储容量，但计算速度较快；算法改进使得地图存储容量小，但计算结果与计算速度不理想。本文结合模型改进和算法改进的特点，提出了一种具有交通规则约束的改进 Dijkstra 算法。

1 问题描述与基本思想

Dijkstra 算法是一种贪心算法，其计算流程从初始节点开始，不断向前搜索，采用标号法将节点分为未访问节点集合和已访问节点集合[9-10]，若某节点被置为已访问状态就再也不会退回到未访问状态，因此可以避免大部分重复计算[11]。而对于实际的交通网络，节点可能存在交通规则约束，需要反复探测该节点是否出现在从初始节点到目的节点的最短路径上。例如图 1 所示的交通网络，若初始节点是 A，目的节点是 G，节点 D 处存在禁止右转的标志，不考虑转向限制的最短路径和次短路径分别是（A，C，D，E，G）和（A，B，D，E，G）。应用传统 Dijkstra 算法向前搜索在第一次访问节点 D 时，由于存在(C，D，E)的转向限制导致节点 D 不会搜索到 E，且节点 D 被置为已访问状态不会被再次访问，使得满足转向限制的最短路径（A，B，D，E，G）不会被发现。从而造成最终计算得到的路径仅仅满足所有转向限制的条件，而未满足最短路径的条件，甚至找不到合适的路径。

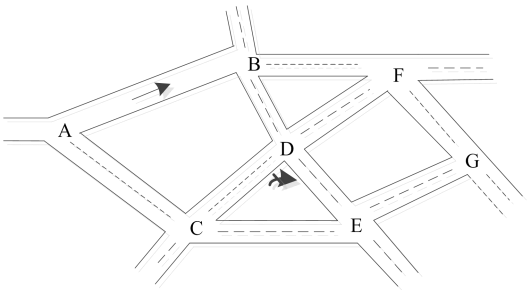


图 1　带转向限制的交通网络

综上所述，传统的 Dijkstra 算法应用于具有交通规则约束的交通网络中时遇到了困难，这里所说的交通规则约束包括禁止掉头、单向行驶、限时行驶、转向限制等。针对此问题，本文提出一种基于模型改进和算法改进的 Dijkstra 算法。其基本思想是：

(1) 对于禁止掉头约束。只需在路网模型的相应路段设置标志(σ_{nt}^{e})，记录该约束条件，路径规划仅仅从车辆当前行驶路段终点开始，而另一端点不予以考虑。

(2) 对于单向行驶约束。只需在路网模型中将相应路段抽象为单向边即可。

(3) 对于限时行驶约束。只需在路网模型中相应路段设置限时标志(σ_{tl}^{e})，记录禁行时间，使得在该时间段内该路段对路径规划算法表现为不可见。

(4) 对于转向限制。首先需要在路网模型中具有转向限制的节点处设置转向标志(σ_{dl}^{v})，记录转向约束条件。其次，涉及算法的改进，算法中对节点引入“待选择状态”和“可再更新状态”来辅助解决类似节点 D 的情况，使得该类节点可以被再次访问。利用 father[ ]数组记录各个节点的父节点，同时，增设辅助数 grandfather[ ]，用来记录各个节点的祖父节点，从而保存了（节点，父节点，祖父节点）的三元组信息。

(5) 无约束状态设置标志(σ_{nr}^{ev})。

2 改进的 Dijkstra 算法

2.1 带约束的路网模型

在路径规划之前首先需要进行路网建模，将实际的交通网络转化为抽象模型，得到问题的形式化描述。建模过程中，使用图论中的有向图思想[12]，同时对各路段赋以权值，将路网抽象的表示为有向加权图 G=(V，E，Σ)，其中 V 为所有节点的集合，E 为所有的边的集合，Σ表示节点和边的约束集合 $\{\sigma_{nt}^{e}$，σ_{tl}^{e}，σ_{dl}^{v}，$\sigma_{nr}^{ev}\}$。将单向道路抽象为单向边，双向道路抽象为双向边，交叉口抽象为节点。

$w(e)$ 定义为图 G 中边 e 的权值。设 q 是图 G 中从 s 到 d 的一条路径，且 q 满足所有的交通规则约束，则认为 q 是从 s 到 d 的一条有效路径[13]。将路径 q 中所有的边的权值之和定义为路径的权值，记作 $W(q)$，即：

$$W(q)=\sum_{e\in q}w(e)$$

因此路径规划问题就是在从 s 到 d 的所有有效路径中，找到权值最小的那条路径 q_{min}，即：

$$W(q_{min})=\min_{q\in\chi(s,t)}\{W(q)\}$$

其中，χ(s, t)为从节点 s 到节点 d 的所有有效路径的集合。图 2 为图 1 所示的交通网络所对应的有向加权图，可采用顺序邻接表对有向加权图进行存储[14]。

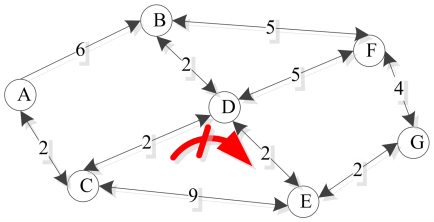


图 2　带约束交通网络的有向加权图

2.2 算法思想

对于禁止掉头约束、单向行驶约束、限时行驶约束可以仅利用上述的改进模型实现，而对于转向限制，还涉及算法的改进。改进算法保存了节点三元组的信息，引入辅助数组

grandfather[ ]，记录各个节点的祖父节点，同时对状态数组 state[ ]做修改，增设“待选择状态”和“可再更新状态”。state 中 0 表示“未访问状态”，1 表示“待选择状态”，2 表示“已访问状态”，3 表示“可再更新状态”。 用 V 表示所有节点的集合，用 N 表示未访问状态节点的集合，用 M 表示待选择节点的集合，用 Q 表示已访问状态节点的集合，用 T 表示可再更新状态节点的集合，s 为初始节点。改进算法的思想描述如下:

(1) 初始化：$dist[s]=0$；$\forall i \in V$且$i \neq s$，令 $dist[i]=+\infty$；$\forall j \in V$，令 $father[j]=-1$，$grandfather[j]=-1$，$state[j]=0$。

(2) 选择访问节点 v：If$(M \neq \varnothing)$，Then 选择 v 使得 $dist[v]=\max_{k \in M} dist_k$，令 $state[v]=2$；Else 选择 v 使得 $dist[v]=\min_{k \in N} dist_k$，令 $state[v]=2$。

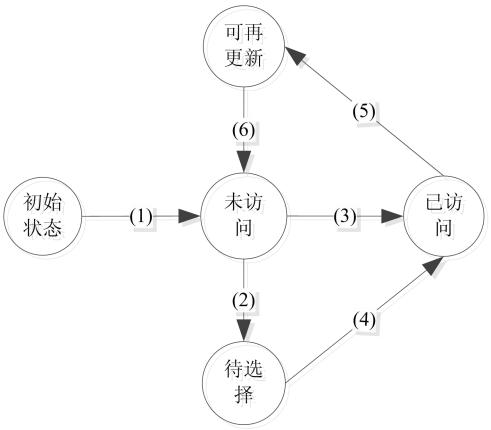
(3) 对于满足如下条件的任意节点 w：
$$\left(\begin{array}{l}\left(w \in \left(adjacent(v) \cap N\right)\right) \&\& \\ \left(dist[v]+weight(<v,w>)<dist[w]\right)\end{array}\right) \| \left(\left(w \in \left(adjacent(v) \cap T\right)\right) \&\& \left(father[w]!=v\right)\right)$$
完成步骤(4)、步骤(5)的操作。

(4) If$\left(\left(father[v],v,w\right) \in \sigma_{dl}^{v}\right)$，
Then 令 $state[v]=3$, goto(3)；
Else If $\left((v,w,X) \in \sigma_{dl}^{w}, \text{其中} X \in adjacent(w)\right)$，
Then 令 $state[w]=1$；
Else 令 $state[w]=0$。

(5) $father[w]=v$；
$dist[w]=dist[v]+weight(\langle v,w \rangle)$；
If$(grandfather[w]=-1)$，
Then $grandfather[w]=father[v]$。

(6) If$\left(\forall j \in V, state[j]==2\right)$，Then return；
Else goto(2)。

其中 adjacent(w)表示 w 的邻接节点组成的集合，节点的状态转换图如图 3 所示。对于步骤(4)，If$\left(\left(father[v],v,w\right) \in \sigma_{dl}^{v}\right)$用来判断在节点 v 处是否存在以 v 的父节点开始经过 v 到达节点 w 的转向限制；If$\left((v,w,X) \in \sigma_{dl}^{w}, \text{其中} X \in adjacent(w)\right)$用来判断在节点 w 处是否存在以 v 开始的经过 w 到达 w 的任意邻接节点的转向限制。



(1)初始化，令其state=0；
(2)存在从其父节点开始经该节点到其任意子节点的转向限制；
(3)其dist值为未访问节点集中最小的；
(4)其dist值为待选择节点集中最大的；
(5)存在从其父节点开始经该节点到其任意子节点的转向限制；
(6)该节点被再次更新。

图 3　节点状态转换图

经过上述处理便可以得到交通网络的最终路径规划表，通过回溯策略可以计算除带有转向限制的节点到初始节点的最短路径，回溯策略的思想是：从目的节点开始，将其与其父节点的存入结果队列；自此针对路径上的各个节点 j，查看其父节点处是否存在任意方向上的转向限制，如果不存在则按其父节点 father[j]回溯，否则按其子节点 i 的祖父节点 grandfather[i]回溯；直到到达初始节点为止，结果队列中的节点组成的路径即为从初始节点到目的节点的符合转向限制的最短路径，其流程图如图 4 所示。而对于带有转向限制的节点处的路径规划问题，可以在路径规划过程中第一次到达该节点时终止搜索，接着按回溯策略便可得到从初始节点到该节点的最短路径。

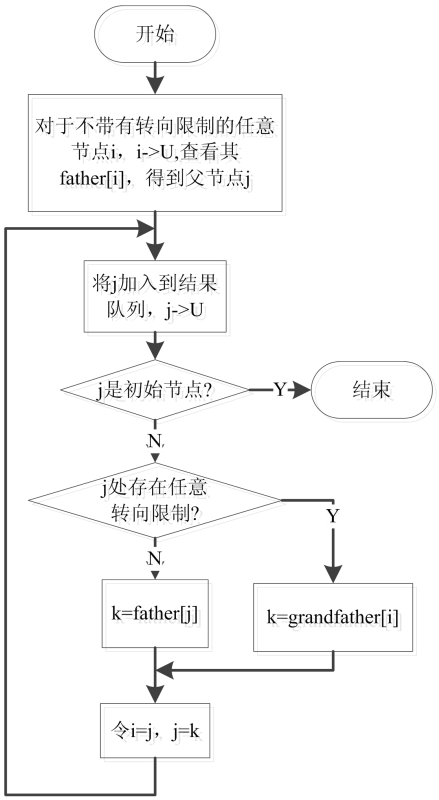


图 4 回溯流程图

2.3 算法分析

(1) 时间复杂度。对于具有 n 个节点的交通网络，其交通规则约束集合为：

$$Rule=\left\{Rule(i)\,|\,1\le i\le n\right\}$$

其中 $Rule(i)$ 表示节点 i 的交通规则约束集，其交通约束的数量为 r_i，因此整个网络的交通规则约束的数量为:

$$r=\sum_{i=1}^{n}r_i$$

那么对于节点 i，最多搜索次数为 r_i+1，因此算法在最坏情况下的循环次数为:

$$\sum_{i=1}^{n}(r_i+1)=r+n$$

与文献[7]的搜索次数相同。在循环内部，对于节点 i，其邻接节点个数为 a_i，在最坏情况下 a_i+1，所以循环内部的时间复杂度为 $O(n-1)=O(n)$。对于一般的交通网络，r 不会超过一个常数，故本文算法在最坏情况下的时间复杂度为:

$$O((r+n)*n)=O(\mathrm{n}^2)$$

(2) 空间复杂度。

本文为每个节点分配 4 个存储空间，分别为状态(state)、路径长度(dist)、父节点(father)、祖父节点(grandfather)，用来记录路径规划过程中的各种信息，对于具有 n 个节点的交通网络需要 4n 的存储空间，故本文算法的空间复杂度为 $O(n)$，相对于文献[7]有较好的空间复杂度，特别适合用于解决嵌入式环境下的路径规划问题。

2.4 算法证明

下面用 R 表示带有转向限制的节点集合。对于不存在转向限制的交通网络，改进算法退化为 Dijkstra 算法，由 Dijkstra 算法的正确性，保证了改进算法的正确性。

对于存在转向限制的交通网络，采用数学归纳法，用下列断言作为归纳假设：(其中 s 为初始节点)

在第 k 次迭代中

(1) 在 Q-R 中的任意节点 i，其 dist 值是从 s 到 i 的满足转向限制的最短路径的长度。

(2) 在 N-R 中除 dist 值为+∞的任意节点 j，其 dist 值是满足转向限制的从 s 到 j 的最短路径的长度，且该路径只包含 Q+R 中的节点。

当 k=0 时，没执行任何迭代之前，Q={s}，很显然断言(1)，(2)是正确的，因此基本情况是正确的。

假定归纳假设对第 k（k>0）次迭代成立。令 v 是第 k+1 次迭代添加到 Q 中的节点，使得 v 是在第 k 次迭代结束时在 M 中 dist 值最大的节点或在 N 中 dist 值最小的节点。（在该节点不是唯一的情况下，可以任意选择）。

根据归纳假设，可得在第 k+1 次迭代之前，在 Q-R 中的任意节点，它的 dist 值都是从 s 出发的到该节点最短路径的长度。下面根据节点 v 处是否存在转向限制分情况讨论：(a)节点 v 处不存在转向限制，$dist_{k+1}[v]$ 一定是从 s 到 v 的最短路径长度，否则，在第 k 次迭代后，就可能存在长度小于 $dist_k[v]$ 的从 s 到 v 路径，该路径包含不在 Q+R 中的节点（因 $dist_k[v]$ 是在第 k 次迭代后，只包含 Q+R 中节点的从 s 到 v 的最短路径的长度）。设 u 是在该路径里不在 Q+R 的第一个节点，则存在一条从 s 到 u 的只包含 Q+R 中节点且长度小于 $dist_k[v]$ 的路径，这与对节点 v 的选择相矛盾，因此在第 k+1 次迭代结束时断言 1)在该情况下成立。(b)节点 v 处存在转向限制，则 $v\in R$，故必然不属于 Q-R。因此该情况不影响断言(1)的正确性。综上所述，在第 k+1 次迭代结束时断言(1)成立。

设 u 是算法在第 k+1 次迭代后不属于 Q+R 的任意节点。下面根据节点 v 处是否存在转向限制分情况讨论：(a)节点 v 处不存在转向限制，从 s 到 u 的只包含 Q+R 中节点的最短路径或者包含 v，或者不包含 v。若不包含 v，根据归纳假设它的长度 $dist_{k+1}[u]=dist_k[u]$。若包含 v，则它必然这样

组成：一条从 s 到 v 的具有最短可能长度的路径，其中还包含了 Q+R 中不同于 v 的元素，后面接着从 v 到 u 的边，在这种情况下 $dist_{k+1}[u]=dist_k[v]+weight(\langle v,u\rangle)$。故该情况下断言(2)为真，因为

$$dist_{k+1}[u]=\min\begin{Bmatrix}dist_k[u],\\ dist_k[v]+weight(\langle v,u\rangle)\end{Bmatrix}。\quad (b)$$

节点 v 处存在转向限制，节点 u 或者在该转向限制上，或者不在该转向限制上。若不在该转向限制之上，则该情况与(a)所述情况相同，不再累述。若在该转向限制之上，则从 s 到 u 的只包含 Q+R 中节点的最短路径不会包含 v，根据归纳假设它的长度 $dist_{k+1}[u]=dist_k[u]$，故该情况下断言(2)也为真。综上所述，在第 k+1 次迭代结束时断言(2)成立。

证毕。

2.5 算法应用

采用本文的算法对图 5 所示的交通网络有向加权图进行处理，每步循环后各个数组的变化情况如图 6 所示。其中在该路网的节点 E、F、H 处分别存在(D，E，I)、(E，F，J)、(D，H，I)的转向限制，如图 5 所示。

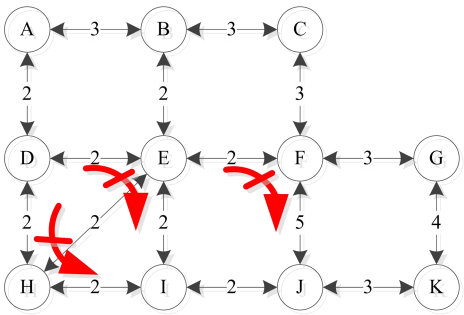


图 5　带多个约束交通网络的有向加权图

初始化	A	B	C	D	E	F	G	H	I	J	K
s	**2**	0	0	0	0	0	0	0	0	0	0
d	**0**	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
f	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
g	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

扩展A	A	B	C	D	E	F	G	H	I	J	K
s	2	0	0	**2**	0	0	0	0	0	0	0
d	**0**	**3**	∞	**2**	∞	∞	∞	∞	∞	∞	∞
f	-1	**A**	-1	**A**	-1	-1	-1	-1	-1	-1	-1
g	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

扩展D	A	B	C	D	E	F	G	H	I	J	K
s	2	0	0	2	**1**	0	0	**2**	0	0	0
d	0	3	∞	2	**4**	∞	∞	**4**	∞	∞	∞
f	-1	A	-1	A	**D**	-1	-1	**D**	-1	-1	-1
g	-1	-1	-1	-1	**A**	-1	-1	**A**	-1	-1	-1

扩展H	A	B	C	D	E	F	G	H	I	J	K
s	2	0	0	2	**2**	0	0	**3**	0	0	0
d	0	3	∞	2	4	∞	∞	4	∞	∞	∞
f	-1	A	-1	A	D	-1	-1	D	-1	-1	-1
g	-1	-1	-1	-1	A	-1	-1	A	-1	-1	-1

扩展E	A	B	C	D	E	F	G	H	I	J	K
s	2	0	0	2	**3**	**2**	0	**0**	0	0	0
d	0	3	∞	2	4	**6**	∞	**6**	∞	∞	∞
f	-1	A	-1	A	D	E	-1	E	-1	-1	-1
g	-1	-1	-1	-1	A	**D**	-1	A	-1	-1	-1

扩展F	A	B	C	D	E	F	G	H	I	J	K
s	2	**2**	0	2	**0**	**3**	0	0	0	0	0
d	0	3	**9**	2	**8**	6	**9**	6	∞	∞	∞
f	-1	A	F	A	F	E	F	E	-1	-1	-1
g	-1	-1	**E**	-1	A	D	**E**	A	-1	-1	-1

扩展B	A	B	C	D	E	F	G	H	I	J	K
s	2	2	0	2	**2**	3	0	0	0	0	0
d	0	3	**6**	2	**5**	6	9	6	∞	∞	∞
f	-1	A	**B**	A	**B**	E	F	E	-1	-1	-1
g	-1	-1	E	-1	A	D	E	A	-1	-1	-1

扩展E	A	B	C	D	E	F	G	H	I	J	K
s	2	2	0	2	2	3	0	**2**	0	0	0
d	0	3	6	2	5	6	9	6	**7**	∞	∞
f	-1	A	B	A	B	E	F	E	**E**	-1	-1
g	-1	-1	E	-1	A	D	E	A	**B**	-1	-1

扩展H	A	B	C	D	E	F	G	H	I	J	K
s	2	2	**2**	2	2	3	0	2	0	0	0
d	0	3	6	2	5	6	9	6	7	∞	∞
f	-1	A	B	A	B	E	F	E	E	-1	-1
g	-1	-1	E	-1	A	D	E	A	B	-1	-1

扩展C	A	B	C	D	E	F	G	H	I	J	K
s	2	2	2	2	2	**0**	0	2	**2**	0	0
d	0	3	6	2	5	**9**	9	6	7	∞	∞
f	-1	A	B	A	B	**C**	F	E	E	-1	-1
g	-1	-1	E	-1	A	D	E	A	B	-1	-1

扩展I	A	B	C	D	E	F	G	H	I	J	K
s	2	2	2	2	2	**2**	0	2	2	0	0
d	0	3	6	2	5	9	9	6	7	**9**	∞
f	-1	A	B	A	B	C	F	E	E	**I**	-1
g	-1	-1	E	-1	A	D	E	A	B	**E**	-1

扩展F	A	B	C	D	E	F	G	H	I	J	K
s	2	2	2	2	2	2	**2**	2	2	0	0
d	0	3	6	2	5	9	9	6	7	9	∞
f	-1	A	B	A	B	C	F	E	E	I	-1
g	-1	-1	E	-1	A	D	E	A	B	E	-1

扩展G	A	B	C	D	E	F	G	H	I	J	K
s	2	2	2	2	2	2	2	2	2	**2**	0
d	0	3	6	2	5	9	9	6	7	9	**13**
f	-1	A	B	A	B	C	F	E	E	I	**G**
g	-1	-1	E	-1	A	D	E	A	B	E	**F**

扩展J	A	B	C	D	E	F	G	H	I	J	K
s	2	2	2	2	2	2	2	2	2	2	**2**
d	0	3	6	2	5	9	9	6	7	9	**12**
f	-1	A	B	A	B	C	F	E	E	I	**J**
g	-1	-1	E	-1	A	D	E	A	B	E	**F**

图 6　各步循环数组更新结果图

该过程中，扩展 D 时发现有(D，E，I)、(D，H，I)的转向限制，将 state[E]、state[H]置为 1，然后从 state 为 1 的节点中选择 dist 值最大作为当前节点，故又将 state[H]置为 2。同理，第一次扩展 E 时对 state[F]也做了相同的处理。从通过最后一张表，便可以得到除带有转向限制的节点到初始节点的最短路径。例如节点 K，其父节点是 J，J 处不存在转向限制，所以 J 的父节点应是 father[J]，即 I；I 处不存在转向限制，所以 I 的父节点应是 father[I]，即 E；E 处存在转向限制，所以 E 的父节点应是 grandfather[I]，即 B；B 处不存在转向限制，所以 B 的父节点应是 father[B]，即 A；由此得到一条从 A 到 K 的最短路径，即(A，B，E，I，J，K)，长度为 12。其他节点也可以按此步骤得到从初始节点到它的最短路径。

3 实验测试

3.1 实验环境

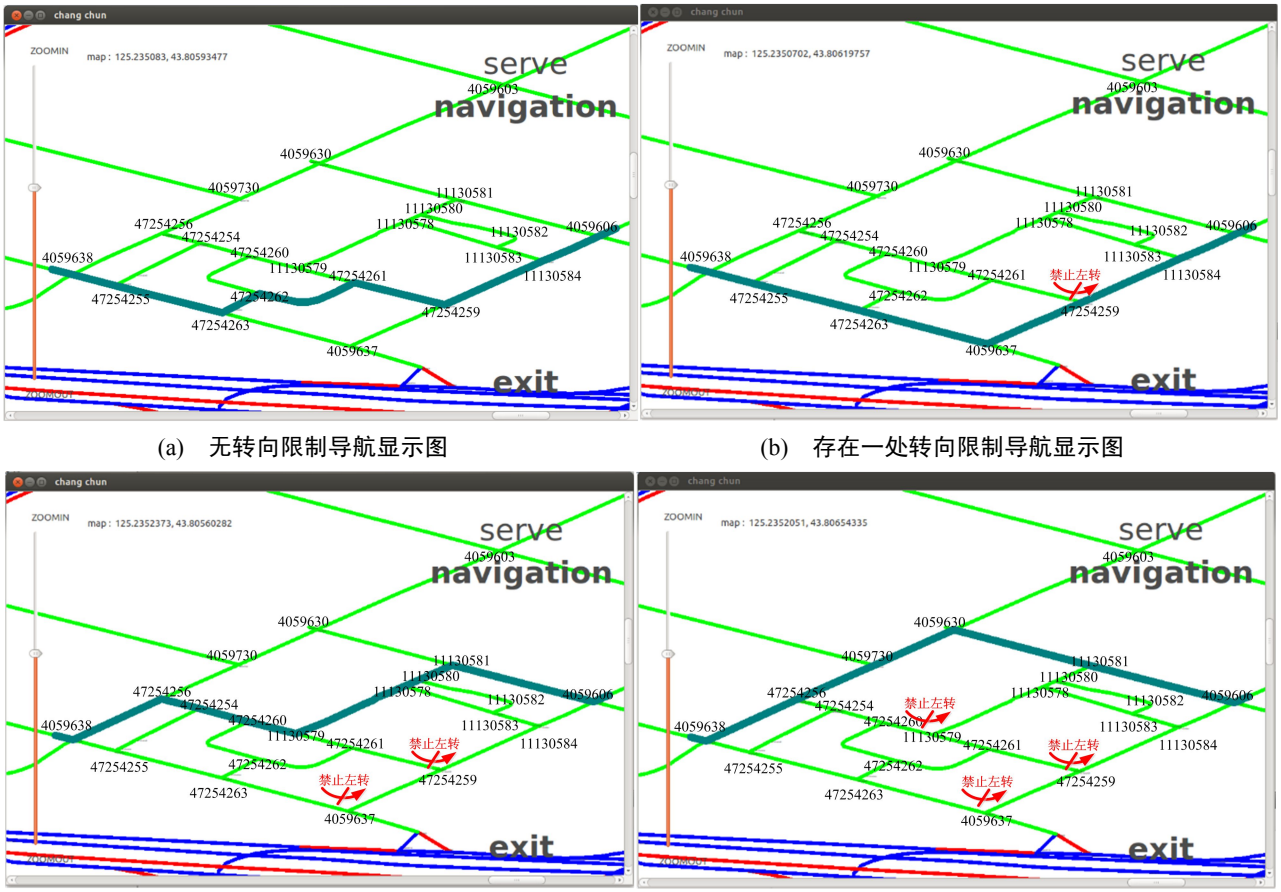
本文依托云平台上的动态轨迹导航系统的研究与开发的项目，在 Linux 操作系统环境下，基于 QT 开发平台对改进算法做了验证。电子地图使用是 MapInfo 的.MID 与.MIF 文件，选取了长春市朝阳区晨光街附近的地图数据。转向限制信息由人为随机加入交通网络。

3.2 结果与分析

在测试实验中，使用改进算法对电子地图以 4059638 为初始节点，以 4059606 为目的节点的需求做最短路径规划。图 7(a)、图 7(b)、图 7(c)、图 7(d)分别为服务器端在无转向限制、存在一处转向限制、存在两处转向限制、存在三处转向限制情况下的导航结果。其中各图中的圆弧箭头代表转向限制。

图 7(a)中无转向限制信息；图 7(b)中，在节点 47254259 处加入了禁止左转的限制；图 7(c)相对于图 7(b)在节点 4059637 加入了禁止左转的限制；图 7(d)相对于图 7(c)在节点 11130579 处加入了禁止左转的限制。

各次路径规划的最短路径长度分别为 588m、594m、596m、598m，由此可见，各次路径规划的最短路径长度呈现递增关系，总能找到考虑交通网络转向限制的最短路径。由此可见，本文提出改进算法能有效解决具有交通规则约束的交通网络的路径规划问题。



(a)　无转向限制导航显示图　　(b)　存在一处转向限制导航显示图

(c)　存在两处转向限制导航显示图　　(d)　存在三处转向限制导航显示图

图 7　服务器端导航显示图

4 结语

对于具有交通规则约束的交通网络，算法通过对节点新增“待选择状态”和“可再更新状态”，引入祖父节点并融合带约束的路网模型，可以计算出交通网络中各节点的邻接节点三元组信息，依据该三元组可以得到从初始节点到目的节点的最短路径，且具有较低的时空复杂度。

参考文献

[1]WANG R, WANG Z. Research on path planning of Intelligent Transport System based on improved bidirectional search[C]//International Asia Conference on Industrial Engineering and Management Innovation (IEMI2012) Proceedings. Berlin Heidelberg: Springer, 2013: 1639-1646.

[2] ZHANG Guanglin, HU Xiaomei, CHAI Jianfei, et al. Summary of path planning algorithm and its application[J]. Modern Machinery, 2011, 37(5): 85-90.(张广林,胡小梅,柴剑飞,等.路径规划算法及其应用综述[J]. 现代机械,2011,37(5):85-90.)

[3]YI Y L, GUAN Y. A path planning method to robot soccer based on Dijkstra algorithm[M]//Advances in Electronic Commerce, Web Application and Communication. Berlin Heidelberg: Springer, 2012: 89-95.

[4]LEI Dongsheng, ZHU Tongyu. A dynamic path planning algorithm based on real-time traffic information[J]. Computer Science,

2008.35(4A): 28-30. (雷东升,诸彤宇.一种基于实时路况信息的动态路径规划算法[J].计算机科学,2008,35(4A): 28-30.)

[5] LIU Jianmei, MA Shoufeng, MA Shuaiqi. Computation method of the dynamic shortest path based on improved-Dijkstra algorithm [J]. Systems Engineering-Theory&Practice, 2011(6): 200-206. (刘建美,马寿峰,马帅奇. 基于改进的 Dijkstra 算法的动态最短路计算方法[J].系统工程理论与实践,2011(6):200-206.)

[6] HLINĚNÝ P, MORIŠ O. Generalized maneuvers in route planning for computing and informatics[J]. Computing and Informatics, 2012, 31(3): 531--549.

[7]YANG Dongkai, CHEN Zhiyu, WU Jinpei, et al. High efficient route planning algorithm based on traffic constraints[J]. Journal of Transportation Systems Engineering and Information Technology, 2008, 8(2): 64-68. (杨东凯,陈志宇,吴今培,等.基于交通约束的高效路径规划算法[J]. 交通运输系统工程与信息,2008,8(2):64-68.)

[8]WANG Q, XU L, QIN J. Research and realization of the optimal path algorithm with complex traffic regulations in GIS[C]//Automation and Logistics, 2007 IEEE International Conference on. Jinan: IEEE, 2007: 516-520.

[9]YAN Banzhong, LIU Jun, ZHANG Bo. Application and simulation of improved Dijkstra algorithm in a vehicle navigation system[J]. Applied Science and Technology, 2012, 38(11): 34-38.(闫保中,刘军,张波.Dijkstra 改进算法在车辆导航系统中的应用与仿真[J].应用科技,2012,38(11):34-38.)

[10] WANG Shuxi. Improved Dijkstra shortest path algorithm and its application[J] . Computer Science, 2012(5): 223-228. (王树西.改进的 Dijkstra 最短路径算法及其应用研究[J].计算机科学,2012(5);223-228.)

[11]FENG Xinxin. Efficient implementation of Dijkstra algorithm in embedded GIS[J].Transactions of Beijing Institute of Technology, 2009, 29(10): 873-876. (冯欣欣.Dijkstra 算法在嵌入式 GIS 中的优化实现[J].北京理工大学学报,2009,29(10):873-876.)

[12]FAN Y, WANG Q, LU D, et al. An improved Dijkstra algorithm used on vehicle optimization route planning[C]// Proceedings of 2010 IEEE Computer Engineering and Technology, Chengdu: IEEE, 2010: 693-696.

[13]LU Kezhong, SUN Hongyuan, LIN Xiaohui, et al.Shortest path algorithm of urban traffic network based on turning restriction[J].Computer Engineering and Applications, 2008, 44(10) : 10- 12.(陆克中,孙宏元,林晓辉,等.一种基于转向限制的城市交通网最短路径算法[J].计算机工程与应用,2009,44(10):10-12.)

[14] ZHENG Xiaotian, QIN Guihe, DONG Jinnan, et al. Design of central navigation system based on COMPASS/GPS dual switch[J].Journal of Jilin University: Science Edition, 2014, 52(5): 989-994.(郑啸天,秦贵和,董劲男,等.基于北斗/GPS 双切换的中心导航系统设计[J].吉林大学;理学版,2014,52(5):989-994.)