

全国第二届部分高校研究生数模竞赛



题目 行车时间估计和最优路线选择 (A)

摘 要:

本题是解决公路车辆行驶时间预测和最优路线选择的问题。

首先,通过对 San Antonio 某段公路一段监测数据进行分析,结合实际情况,提出了两种行车时间估计模型:模型一仅采用各测点的行车速度信息,在单位观测时间内,所有通过测段车辆速度为相邻两测点速度的平均值,车辆行驶时间为测段长度与均值速度的比值,为研究各个测段行驶时间的相互影响,计算了各测段行驶时间向量的相关系数矩阵;模型二综合考虑了观测速度和流量,车辆通过测段的时间由自由通过时间和滞留时间两部分组成。分别采用两种模型对该观测段行车时间进行估测,两种模型计算结果基本吻合,并给出了基于概率的行车时间估计区间;对两种模型进行评价,提出了两种模型各自的适用条件及影响因素。

然后,由于各个路段行驶时间的不确定性,现有的交通系统不能准确的估计行驶时间并选择最优路线。为考虑各种路况特征(单个路段平均行驶时间,路段长度和行驶时间方差)对总行车时间的影响,引入广义行车费用,给定各个路况特征对行车时间的影响权重值,进而计算出各个路段的广义行车费用,最优路线即为广义行车总费用最少的路线。特别的,当路段长度的权重值为 1(仅考虑距离影响)时,采用 Dijkstra 算法编制了 VB 程序,可以获得任意两点间的最优路线。另外,结合现有系统提供的信息,根据行车时间估计模型一,可以得到各个路段行驶时间向量的协方差矩阵,进而求出任一行驶路线的平均行驶时间及其方差;比较所有行驶路线的平均行驶时间,得出最优路线,因此,考虑各个路段行驶时间相互影响时,该方法适用于节点较少的最优路线选择问题。

关键词: 行车时间 滞留量 Dijkstra 算法 广义费用 相关性
最优路线

参赛队号 1356

参赛密码 _____
(由组委会填写)

I: 行车时间估计模型

一、问题描述:

公路行车时间估计对于现代交通运输起着重要的作用。为了进行行车时间估计,在美国 San Antonio 的公路上,安装探测装置以测定车流量和车速。探测器每天 24 小时进行数据采集,每 2 分钟采样一次,每次采样时间为 20 秒,每一次采样可以得出当前的车流量和车速,根据某一天下午 3:40 至傍晚 6:58 的测量数据,对以下问题分析研究:

- 1、分析公路的交通特征,即车行通畅情况和交通拥堵情况。
- 2、给出车辆行驶时间的预测模型。

二、问题分析:

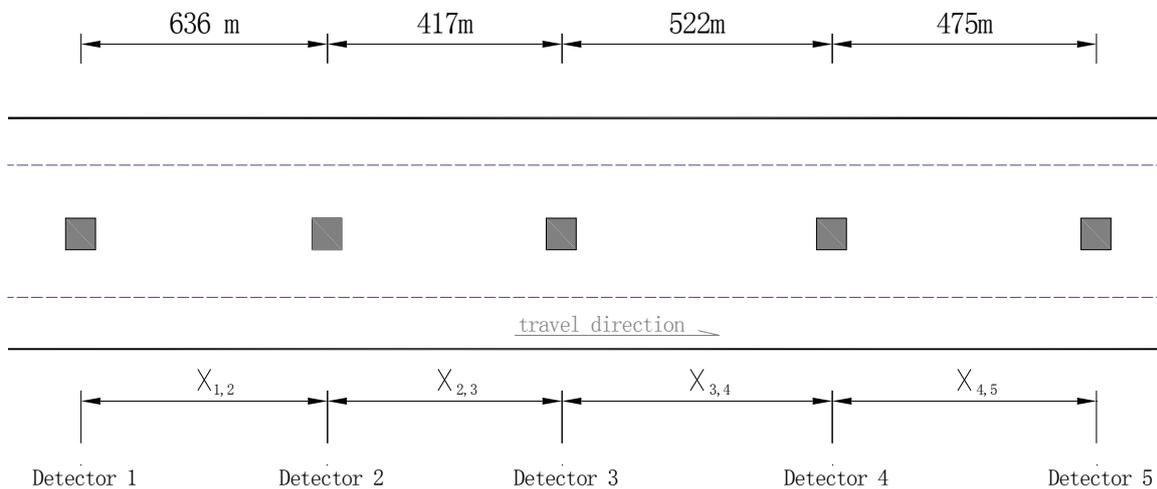


图 1 测点布置图

根据直观了解,如果所有车辆在某一测段内行驶速度变化不大,则相邻两个探测器的测得的速度差值也不会大,因而可以认为探测器所测得的速度值,就是该区间内每辆车的行驶速度,那么行车时间应为该段距离与此速度的商,据此,我们提出第一种车辆行驶时间估计模型,此模型仅与探测器所测得的速度有关。

另外,对于某个测段,如果进入该区间的车辆多于驶出该区间的车辆,其结果就是造成该测段内车辆拥堵,影响正常行驶,则部分车辆将在区间内“耽搁”一些时间,根据这一个实际情况,我们建立了第二种车辆行驶时间估计模型,此模型中考虑了交通流量的影响。

三、模型假设、建立与求解

(一) 第一种行车时间估计模型

1、假设

- (1) 在各路段中车辆单向行驶。
- (2) 测点在 20 秒内的行车状况可以代表 2 分钟内的行车状况。
- (3) 每辆车在两测点间的行驶速度为进入点速度和驶出点处速度的平均值，即：

$$\bar{v} = \frac{1}{2}(v_i + v_j) \quad (1)$$

其中下标 i 表示车辆进入处测点位置， j 表示车辆离开处测点位置。

- (4) 车辆在时刻 t_n 通过各个测段的行驶时间 $T(X_{i,j}, t_n)$ 服从正态分布。

2、模型建立

(1) 基本单元模型

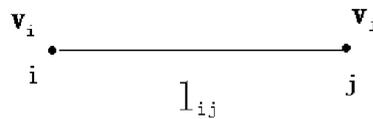


图 2 基本单元模型

车辆通过 l_{ij} 的时间为：

$$T_V = \frac{l_{ij}}{\bar{v}} = 2 \frac{l_{ij}}{v_i + v_j} \quad (2)$$

式中 l_{ij} 表示两测点之间的距离， v_i ， v_j 分别表示驶入测点 i 和驶出测点 j 处的行车速度。

(2) 实用计算模型

根据基本单元模型，可得出 t_n 时刻 $X_{i,j}$ 测段的行车时间为：

$$T_V(X_{i,j}, t_n) = \frac{2l(X_{i,j})}{v(X_i, t_n) + v(X_j, t_n)} \quad (3)$$

式中 i, j —— 分别表示车辆进入点和车辆驶出点；

$X_{i,j}$ —— 表示相邻两探测器间的测量段；

$l(X_{i,j})$ —— 表示 $X_{i,j}$ 测段的长度；

$v(X_i, t_n)$ —— 表示 t_n 时刻，车辆进入 $X_{i,j}$ 测段时的速度；

$v(X_j, t_n)$ ——表示 t_n 时刻，车辆驶出 $X_{i,j}$ 测段时的速度。

则车辆通过 $X_{i,j}$ 测段的平均时间为：

$$T_V(X_{i,j}) = \frac{\sum_{n=1}^N T_V(X_{i,j}, t_n)}{N} \quad (4)$$

式中 N 表示该测段测量数据个数。

3、模型求解

表 1 各测点行车速度(单位: mile/h)

t_n	$v(X_1, t_n)$	$v(X_2, t_n)$	$v(X_3, t_n)$	$v(X_4, t_n)$	$v(X_5, t_n)$
03:40:07 PM	57	54	62	20	58
03:42:07 PM	62	68	63	21	59
.....
06:58:07 PM	64	68	57	36	64

表 2 测段间距(单位: m)

$l(X_{1,2})$	$l(X_{2,3})$	$l(X_{3,4})$	$l(X_{4,5})$	全程
636	417	522	475	2050

把表 1 和表 2 中的数据代入 (3)、(4) 式，得到各测段行车时间和全程行车时间，结果如下表：

表 3 行车时间及标准差

	$X_{1,2}$	$X_{2,3}$	$X_{3,4}$	$X_{4,5}$	全程
$T_V(X_{i,j})$ (sec)	26.93	26.71	61.55	52.8	167.95
标准差 S (sec)	8.92	40.52	106.84	61.02	165.5

对于不同时刻各测段行车时间和全程（包括四个测段）行车时间，计算结果如图 4、图 5 所示：

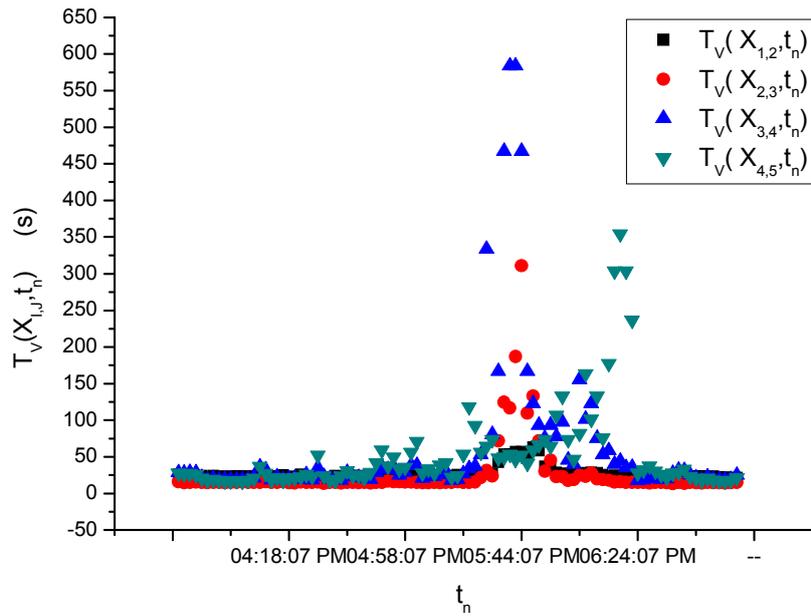


图 4 不同时刻各测段的行车时间

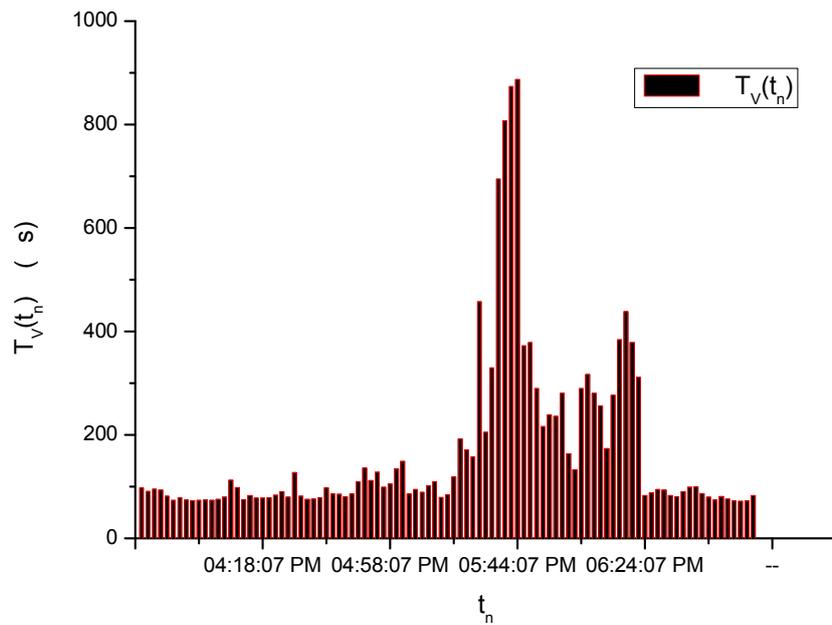


图 5 不同时刻全程的行车时间

4、结果分析及评价

(1) 相关性分析

考虑各个测段行驶时间存在相互影响，用软件 Matlab 求出四个测段行车时间向量

$[T_V(X_{i,j}, t_1) \ T_V(X_{i,j}, t_2) \ \cdots \ T_V(X_{i,j}, t_N)]^T$ 的协方差，协方差矩阵为：

$$C = 10^4 \times \begin{bmatrix} 0.0080 & 0.0310 & 0.0734 & 0.0036 \\ 0.0310 & 0.1642 & 0.3551 & 0.0044 \\ 0.0734 & 0.3551 & 1.1420 & 0.0583 \\ 0.0036 & 0.044 & 0.0583 & 0.3723 \end{bmatrix}$$

用函数 $CORRCOEF(X)$ 函数求得四个测段行车时间的相关系数，其相关系数矩阵为：

$$CF = \begin{bmatrix} 1.0000 & 0.8589 & 0.7703 & 0.0662 \\ 0.8589 & 1.0000 & 0.8203 & 0.0177 \\ 0.7703 & 0.8203 & 1.0000 & 0.0895 \\ 0.0662 & 0.0177 & 0.0895 & 1.0000 \end{bmatrix}$$

矩阵 CF 中的元素 CF_{ij} ，其中 $CF_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} \times C_{jj}}}$ 。

由相关函数矩阵可以看出， $T_V(X_{1,2})$ 与 $T_V(X_{2,3})$ 、 $T_V(X_{3,4})$ 的相关系数分别为 0.8589 和 0.7703，说明 $T_V(X_{1,2})$ 与 $T_V(X_{2,3})$ 、 $T_V(X_{3,4})$ 之间相互影响程度较大；而 $T_V(X_{4,5})$ 与 $T_V(X_{1,2})$ 、 $T_V(X_{2,3})$ 、 $T_V(X_{3,4})$ 的相关系数则分别为 0.0662、0.0177、0.0895，其值均较小，这说明 $T_V(X_{4,5})$ 对 $T_V(X_{1,2})$ 、 $T_V(X_{2,3})$ 、 $T_V(X_{3,4})$ 的影响较小。四个测段的行车时间中，最大相关系数 0.8589 出现在 $T_V(X_{1,2})$ 与 $T_V(X_{2,3})$ 之间，说明该两测段行车时间的相关程度最大，在进行行车时间评估时应对此重点进行考虑。

(3) 模型评价及建议

从图 4、图 5 可以看出，在特定交通时段内，其行车时间明显高于其它时段的行车时间；另一方面，由表 3 计算结果也看出，如果对整个观测时段得到的全部数据一起进行统计分析，其均值标准差 S 过大。故结合实际情况，建议将交通时段分成两部分，交通高峰期和非交通高峰期。对图 4、图 5 进行观测分析，把 05:15 PM ~06:25PM 定为交通高峰期，其余时间定为非交通高峰期。

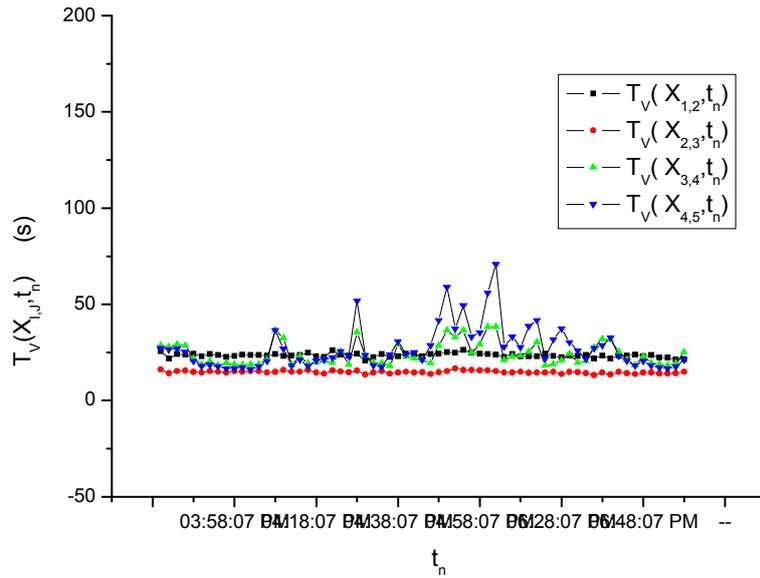


图 5 非高峰期各测段的行车时间

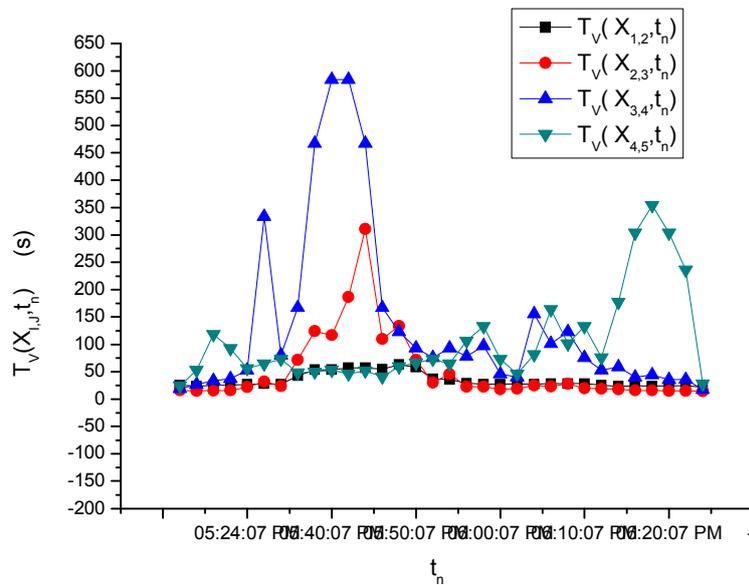


图 6 非高峰期各测段的行车时间

表 4 高峰期和非高峰期各测段行车时间和全程行车时间

		$X_{1,2}$	$X_{2,3}$	$X_{3,4}$	$X_{4,5}$	全程
非高峰期	$T_V(X_{i,j})$ (sec)	23.5	14.8	24.1	27.2	89.5
	标准差 S (sec)	1.0	0.7	5.9	11.2	17.0
高峰期	$T_V(X_{i,j})$ (sec)	32.5	44.2	108.0	108.4	293.0
	标准差 S (sec)	12.0	60.0	115.1	84.6	222.1

在非高峰期交通时段，行车时间按 $(\mu - 2\sigma, \mu + 2\sigma)$ 进行估计，其中 μ 为均值， σ 为

方差，则从第 1 测点到第 5 测点间的行车时间的估计区间（55.5 秒，129.5 秒），保证概率为 95.44%。在高峰期交通时段，第 1 测点到第 5 测点间行车时间数据离散性较大，给出行车时间的预测区间意义不大，故只给出平均行车时间：293 秒。

（二）第二种时间估计模型

1、模型假设

- （1）在各路段中车辆单向行驶。
- （2）测点在 20 秒内的行车状况可以代表 2 分钟内的行车状况。

$$Q(X_i, t_n) = 6 \times q(X_i, t_n) \quad , \quad Q(X_j, t_n) = 6 \times q(X_j, t_n) \quad (5)$$

式中 $q(X_i, t_n)$ ——20 秒内进入测段 $X_{i,j}$ 的车辆数；

$q(X_j, t_n)$ ——20 秒内驶出测段 $X_{i,j}$ 的车辆数；

$Q(X_i, t_n)$ ——120 秒内进入测段 $X_{i,j}$ 的车辆数；

$Q(X_j, t_n)$ ——120 秒内驶出测段 $X_{i,j}$ 的车辆数。

- （3）车辆在测段间的行驶时间为自由通过时间 T_F 和滞留时间 T_C 的代数和，且 T_F 和 T_C 不相关，即：

$$T_T = T_F + T_C \quad (6)$$

- （4）车辆自由通过某测段的速度 $v_f(X_{i,j})$ 为两测点间所测得的平均行车速度最大值，即：

$$v_f(X_{i,j}) = \text{Max} \left\{ \frac{v(X_i, t_n) + v(X_j, t_n)}{2}, \forall n = 1, 2, \dots, N \right\} \quad (7)$$

- （5）车辆在某测段内的滞留时间 T_C 只与该测段内的滞留量有关。

- （6）滞留量不能为负，当进入车辆数小于驶出数量时，认为没有滞留车辆，滞留量为零。

2、模型建立

（1）自由通过时间计算模型

根据假设（4），车辆在 $X_{i,j}$ 测段的自由通过时间为：

$$T_F(X_{i,j}) = \frac{l(X_{i,j})}{v_f(X_{i,j})} \quad (8)$$

(2) 滞留时间模型

1) 滞留时间基本模型

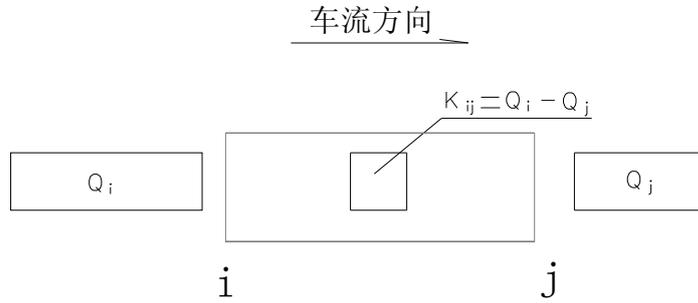


图 7 测段内车辆滞留量图

Q_i 表示 2 分钟内进入测点 i 的车辆数, Q_j 表示 2 分钟内驶出测点 j 的车辆数, 则 $K_{i,j}$ 表示该 2 分钟内滞留在 $X_{i,j}$ 段内的车辆数, 根据假设 (6), 车辆滞留量不能为负, 所以对 $K_{i,j}$ 定义如下:

$$K_{i,j} = \begin{cases} Q_i - Q_j & Q_i - Q_j > 0 \\ 0 & Q_i - Q_j \leq 0 \end{cases} \quad (9)$$

从统计意义上来讲, 滞留在 $X_{i,j}$ 测段内的车辆都被“耽搁”2 分钟, 则在 $X_{i,j}$ 内, 由于交通拥堵“耽搁”的总时间量为 $T'_C = \Delta t \times K_{i,j}$, 其中 Δt 为 2 分钟, 即 120 秒。对于“通过”该测段的每辆车来说, 在该测段因为滞留而被“耽搁”的平均时间为:

$$T_C = \frac{T'_C}{Q_j} = \frac{\Delta t \times K_{i,j}}{Q_j} \quad (10)$$

2) 滞留时间实用计算模型

基本模型中 $K_{i,j}$ 没有考虑 t_{n-1} 时刻滞留的车辆在 t_n 时刻影响, 如果用 t_{n-1} 时刻滞留量与 t_n 时刻车辆滞留量两者的平均值作为在 $X_{i,j}$ 测段内在 t_n 时刻的车辆滞留量, 更为合理。

故将 $K_{i,j}$ 修正为 $K'_{i,j}$:

$$K'(X_{i,j}, t_n) = \frac{K(X_{i,j}, t_{n-1}) + K(X_{i,j}, t_n)}{2} \quad (11)$$

$$K(X_{i,j}, t_n) = \begin{cases} Q(X_i, t_n) - Q(X_j, t_n) & Q(X_i, t_n) - Q(X_j, t_n) > 0 \\ 0 & Q(X_i, t_n) - Q(X_j, t_n) \leq 0 \end{cases} \quad (12)$$

由 (10)、(11) 式可以得出在测段 $X_{i,j}$ 内的滞留时间:

$$T_C(X_{i,j}, t_n) = \frac{\Delta t \cdot K'(X_{i,j}, t_n)}{Q(X_j, t_n)} = \frac{K(X_{i,j}, t_{n-1}) + K(X_{i,j}, t_n)}{2Q(X_j, t_n)} \cdot \Delta t \quad (13)$$

式中 $T_C(X_{i,j}, t_n)$ ——在 t_n 时间段内每辆通过车辆在测段 $X_{i,j}$ 上的滞留时间;

Δt ——两次测量的时间间隔, 为 120 秒。

则车辆通过 $X_{i,j}$ 段的平均滞留时间为:

$$T_C(X_{i,j}) = \frac{\sum_{n=1}^N T_C(X_{i,j}, t_n)}{N} \quad (14)$$

式中 N 表示该测段测量数据个数。

3、模型求解

表 6 各测点行车速度和车流量

t_n	Detector 1		Detector 2		Detector 3		Detector 4		Detector 5	
	Speed (mile/h)	Flow								
03:40:07 PM	57	10	54	9.7	62	8.9	20	10.7	58	5.9
03:42:07 PM	62	9.5	68	11.4	63	13.6	21	10	59	12.2
.....
06:58:07 PM	64	4.2	68	3.9	57	4.5	36	4.4	64	4.1

Speed : $v(X_i, t_n)$ Flow : $q(X_i, t_n)$

把表 6 数据代入 (5)、(12)、(13)、(14) 式, 计算滞留时间 $T_C(X_{i,j})$; 代入 (7)、(8) 式计算自由通过时间 $T_F(X_{i,j})$; 把 $T_C(X_{i,j})$, $T_F(X_{i,j})$ 的计算值代入 (6) 式得到行车总时间 $T_T(X_{i,j})$ 。 $T_T(X_{i,j})$ 计算结果如表 7 所示, 按照该模型计算, 走完全程需要的平均时间估计值为 166.95 秒。

表 7 探测器所测各测点行车速度和车流量

	$X_{1,2}$	$X_{2,3}$	$X_{3,4}$	$X_{4,5}$	全程
$l(X_{i,j})$ (m)	636	417	522	475	2050
$v_f(X_{i,j})$ (m/s)	30.62	31.51	29.95	29.27	—
$T_F(X_{i,j})$ (s)	20.77	13.23	17.43	16.23	67.66
$T_C(X_{i,j})$ (s)	31.63	20.61	35.99	11.06	99.29
$T_T(X_{i,j})$ (s)	52.40	33.84	53.42	27.29	166.95

四、两种行车时间估计模型评价与改进

第一种时间估计模型和第二种时间估计模型都假设了 20 秒内的行车状况能代表 2 分钟内的行车状况，在第一模型中，只考虑了行车速度影响因素，而此行车速度为 20 秒内的行车速度，显然，如果增加观测时间，这个模型的精度将提高。对于第二模型，考虑了行车速度和流量的影响，在这一假设条件下，降低了流量的测量精度。

表 8 两种模型行车时间计算结果比较

	$T(X_{1,2})$	$T(X_{2,3})$	$T(X_{3,4})$	$T(X_{4,5})$	全程时间
第一种模型	26.93	26.71	61.55	52.8	167.95
第二种模型	52.40	33.84	53.42	27.29	166.95

第一种时间估计模型计算的全程行车时间为 167.95 秒，第二种时间估计模型计算的全程时间预计值为 166.95 秒，两者计算值基本相同；但是，由表 8 可以看出，对于各个测量段，所计算出的行车时间估计值有一定差别。从全程 2.050 公里来看，可以认为流量对行车时间估计没有影响，但是对于单个测段，流量对行车时间的估计值有一定影响。

第一种模型仅考虑了测点的行车速度，适用于距离较短、交通状况良好的情况下的行车时间估计，第二种模型综合考虑了最大行车速度和流量对行车时间的影响，当各测段内流量发生变化时，认为交通状况变化（例如：交通拥堵）产生车辆滞留，适用于测段距离较长的路段行车时间估计。但该模型为一种简化模型，它的第（6）个假设认为滞留量不能为负，这一假设有一定的局限性，实际情况应考虑观测区间长度和观测时间间隔长度的影响，当测段长度比较大，或者观测时间间隔比较小时，滞留量可以为负，对于这类情况该模型有待改进。

II&III: 最优路线的选择和行驶时间评估模型

一、问题描述

由于各个路段行驶时间的不确定性, San Antonio 现有的交通系统在提供最优路线和可靠的行驶时间估计时存在缺陷。现有的交通系统中的路况信息包括: 路段正常行驶, 路段拥堵, 路段严重拥堵, 路段在施工(但车辆运行情况不明), 实时信息牌(某些信息牌在某些时刻没有路况信息显示)在各个路段的分布情况。这些路况信息每隔 5 分钟更新一次, 用鼠标点击某一路段, 还可以获得该路段的运行速度信息。由于各个路段行驶时间的不确定性, 现有的系统不能准确的给出最优路线和可靠的时间估计值。因此, 需要给出各路段行驶时间不确定性的衡量指标, 并在此基础上建立合理的时间评估模型和最优路线选择模型。

二、问题分析

通常, 在每条可供选择的路线的特征确定之后, 最优路线的选择还要根据司机的行驶习惯来确定。也就是说, 最优路线的确定要满足多目标的约束(比如行驶时间、路程、是否堵车等各个方面的因素), 还要满足司机在选择路线时的个人习惯。这样, 建立每一种类型的司机行驶时间(司机从出发地到达目的地的时间消耗)的数学表达式是比较困难的, 而对这样一个多因素影响下的行驶时间函数, 其最优解的确定变成了求解 NP 问题, 这也是相当困难的。

根据以上分析, 最优路线选择时需要考虑的问题包括:

- 1、考虑各种因素的影响, 直接计算出发地到目的地之间的时间(NP 问题)计算耗时很长, 不适合大规模交通系统使用。
- 2、个别路段不能提供相关路况信息或信息不准确。
- 4、不同司机选择最优路线的决策方法不同。

三、模型假设、建立与求解

(一) 不考虑各个路段行驶时间的相关性

1、假设

- (1)各个路段的行驶时间 T 都服从正态分布。
- (2)假定利用现有系统可以获得的各个路段的路况, 采用问题 I 中提出的方法, 可以得到在时刻 t 通过各个路段的平均时间 $T(t)$, 以及行驶时间的均方差 $\sigma_T(t)$ 。
- (3)各个路段的平均行驶时间 $T(t)$, 各个路段行驶时间的方差 $\sigma_T(t)$ 都是相互独立的随机变量。

(4)司机对各种路况特征的使用偏好为已知。

(5)以各个路段的平均行驶时间 $T(t)$ ，行驶时间的方差 $\sigma_T(t)$ 和路段的长度作为表示路段特征的指标。

(6)在司机通过任意到达目的地路线的时间 Δt 内，各个路段的路况特征指标不变。

(7)在各路段中车辆单向行驶。

2、建立模型

根据以上假设和分析，可以建立第 i 个路段上广义行驶费用的线性方程：

$$Y_i = \sum_{j=1}^3 w_j Z_{ij}(t) = w_1 Z_{i1}(t) + w_2 Z_{i2}(t) + w_3 Z_{i3}(t) \quad (15)$$

式中： w_j 表示路段第 j 个特征指标在广义行驶费用中占的权重；

$Z_{ij}(t)$ 表示 t 时刻第 i 个路段上第 j 个特征指标的标准化值（即第 i 个路段第 j 个特征值与所有路段第 j 个特征值最大值的比值），根据假设(6)，

$$Z_{ij}(t) = Z_{ij}(t + \Delta t)；$$

$Z_{i1}(t)$ 表示 t 时刻第 i 个路段平均行驶时间的标准化值（ $0 < Z_{i1}(t) \leq 1$ ）

$Z_{i2}(t)$ 表示 t 时刻第 i 个路段长度的标准化值（ $0 < Z_{i2}(t) \leq 1$ ）

$Z_{i3}(t)$ 表示 t 时刻第 i 个路段平均行驶时间方差的标准化值（ $0 < Z_{i3}(t) \leq 1$ ）

$Y_i(t)$ 表示 t 时刻第 i 个路段上行驶需要消耗的广义费用。

显然在计算广义费用时，还可以考虑更多的因素，如路段上是否施工，出口数目，是否有实时信息牌等，这些都可以作为反映路段状况特征的指标。

考虑司机的偏好，可以输入相应的权重值，例如 $w_1 = 0.5, w_2 = 0.3, w_3 = 0.2$ 。特别的，当 $w_1 = 1$ 时，即司机仅以平均行驶时间作为在该路段行驶的广义费用；而 $w_2 = 1$ 时，即司机仅以路段长度作为行驶费用的衡量指标。

对于本题而言，可以根据解决问题 I 的方法，计算各个路段 t 时刻的平均行驶时间 $T(t)$ 以及行驶时间的均方差 $\sigma_T(t)$ ，各个路段的长度已知。司机给定一组权重值，把已知数据带入(15)式就可以得到在 t 时刻该司机出发通过各个路段需要花费的广义费用 $Y_i(t)$ 。

这样，最优路线的选择就变为确定到达目的地消耗广义费用最少的路线，即最有路

线应当满足:

$$Y(t) = \min(\sum_j Y_i(t)) \quad (16)$$

式中: j 表示第 j 条可以到达目的地的路线($j = 1, 2, \dots$), 路线 j 需经过的路段包括 ($i = 1, 2, \dots$)。

那么, 最优路线的选择问题就转化为类似求解最短路径的问题, 可以采用图论中的 Dijkstra 算法求解。

3、模型求解

Dijkstra 算法是典型最短路算法, 用于计算一个节点到其他所有节点的最短路径。主要特点是以起始点为中心向外层层扩展, 直到扩展到终点为止。其算法描述如下:

(1) 初始化集合 E , 使之只包含源节点 S , 并初始化集合 R , 使之包含所有其它节点。初始化路径列 O , 使其包含一段从 S 起始的路径。这些路径的长度值等于相应链路的量度值, 并以递增顺序排列列表 O ;

(2) 若列表 O 为空, 或者 O 中第 1 个路径长度为无穷大, 则将 R 中所有剩余节点标注为不可达, 并终止算法;

(3) 首先寻找列表 O 中的最短路径 P , 从 O 中删除 P 。设 V 为 P 的最终节点。若 V 已在集合 E 中, 继续执行步骤 2。否则, P 为通往 V 的最短路径。将 V 从 R 移至 E ;

(4) 建立一个与 P 相连并从 V 开始的所有链路构成的候选路径集合。这些路径的长度是 P 的长度加上与 P 相连的长度。将这些新的链路插入有序表 O 中, 并放置在其长度所对应的等级上。继续执行步骤 2。

本文针对 $w_1 = 0, w_2 = 1, w_3 = 0$, 即考虑司机仅以路段长度最为衡量广义费用的指标, 根据 Dijkstra 算法, 用 VB 语言编制了取 $w_2 = 1$ 时, 求解任意两点间最优路线的程序 Optimal Route.exe。编制程序时, 在数据文件 (data.txt) 中给出各个节点的信息, 这些信息包括: 节点名、与该节点直接相连接的其它节点号、相连节点间的距离。

图 8 表示 San Antonio 的交通网点, 图中共有 14 个节点, 每两个相邻节点间的距离均已表示在图中。车辆可以通过某一节点到达与该节点相邻的下一路段。

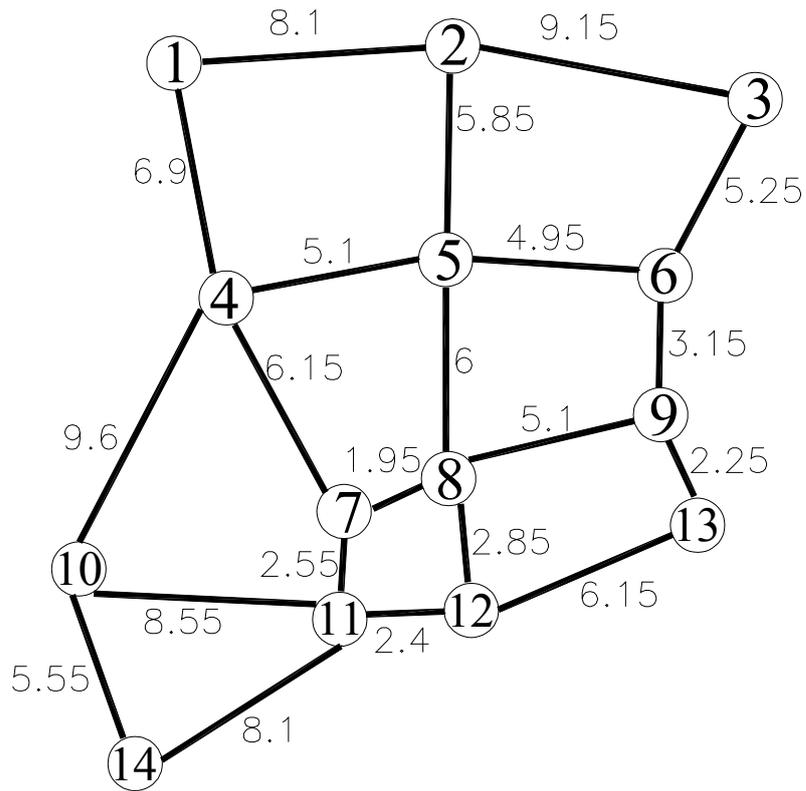


图 8 最优路径选择计算简化图（单位：英里）

利用本文给出的程序可在图 8 的 14 个节点中，进行任意两个节点间最优路线的选择，即给出起始点和目的地，程序自动给出车辆行驶的最短路径。程序界面见图 9。

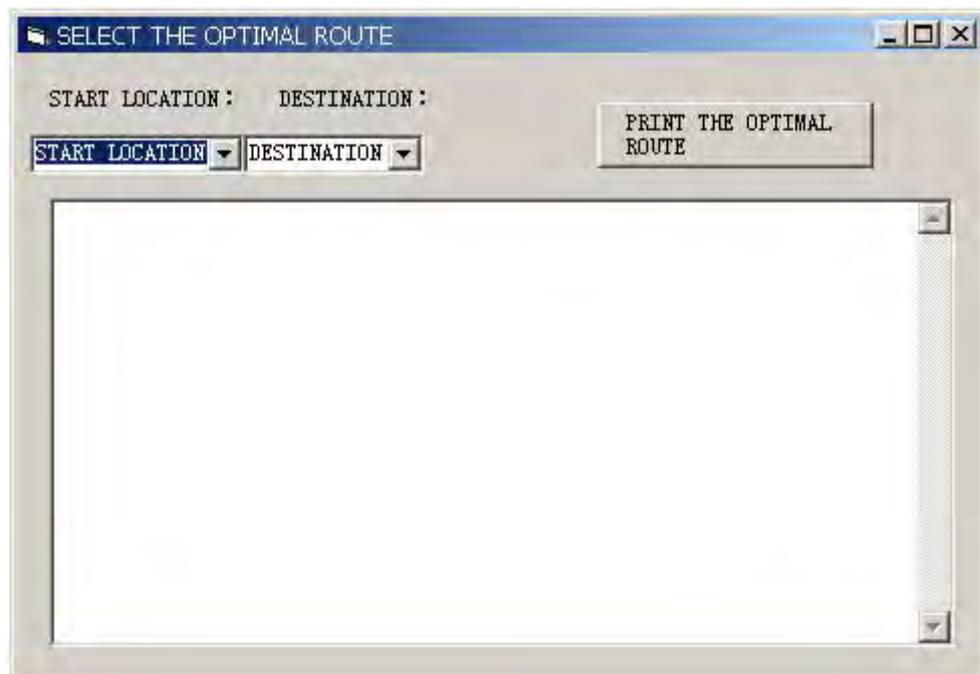


图 9 最优路径计算程序界面

如在节点 3 与节点 14 之间寻找最优路径：在“START LOCATION”中选择“ST03”，

表示起始点为节点 3；在“DESTINATION”中选择“ST14”，表示目的地为节点 14，点击按钮“PRINT THE OPTIMAL ROUTE”即可以得出节点 3 与节点 14 之间的最优路径。由图 10 可见，该最优路径为：节点 3→节点 6→节点 9→节点 8→节点 7→节点 11→节点 14。全路程长度为 26.1 英里。由于只考虑车辆单向行驶，故由节点 14 至节点 3 之间的最优路径也与此相同，即为：节点 14→节点 11→节点 7→节点 8→节点 9→节点 6→节点 3。

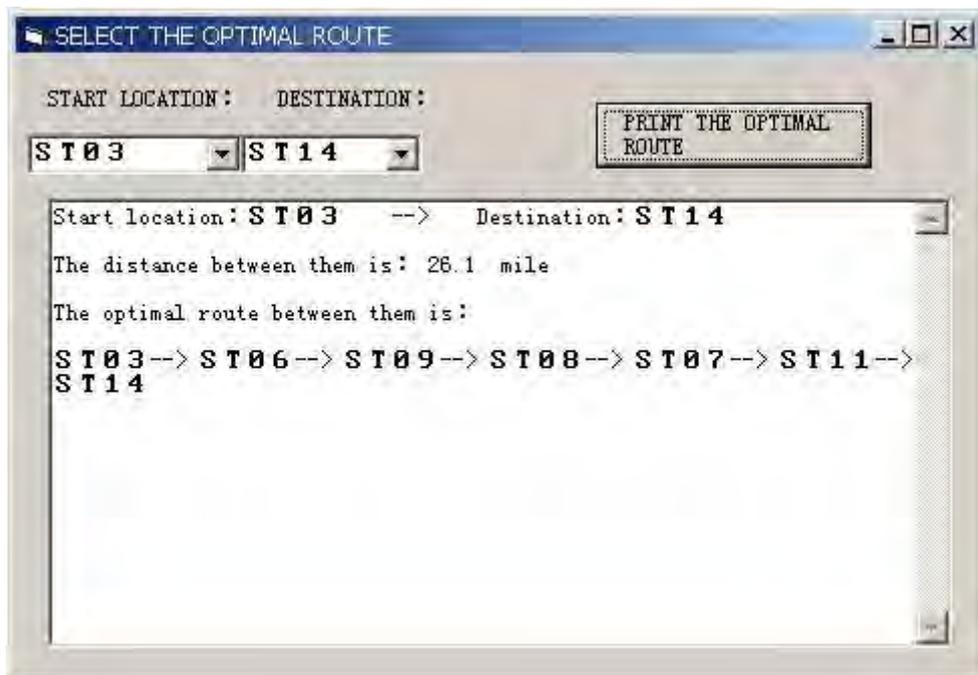


图 10 节点 3 与节点 14 间的最优路径选择

(二) 考虑各个路段行驶时间的相关性

考虑更加接近实际的情况，各个路段上的行驶时间实际上是互相关的随机变量。而协方差矩阵则反映了这些互相关联的随机变量之间的相关程度。不失一般性，假定一条从出发地到达目的地的路线所包含的路段有 n 个，即有 n 个服从正态分布的随机变量 $T_1(t), T_2(t), \dots, T_n(t)$ 分别表示在 t 时刻各个路段的行驶时间。可以求出 n 维正态随机矢量 $TT(t) = (T_1(t), \dots, T_n(t))$ 的协方差矩阵 C ，则 $TT(t)$ 的联合密度函数可以表示为：

$$p_{TT}(tt) = \frac{1}{(2\pi)^{\frac{n}{2}} |C|^{1/2}} \exp\left[-\frac{1}{2}(tt - m)C^{-1}(tt - m)^T\right] \quad (17)$$

式中： $m = E[TT]$ 和 $C = [c_{ij}]$ 分别是 $TT(t)$ 的均值矢量和协方差矩阵。 $TT(t)$ 服从 $N(m, C)$ 分布。

考虑线性变换 $Y(t) = BTT^T(t)$ ，其中 B 是 $m \times n$ 阶矩阵，则 $Y(t)$ 是 m 维矢量。

根据线性变换的正态不变性，可以求得 $Y(t)$ 的均值矢量和协方差矩阵：

$$m_Y = E[Y] = Bm^T, \quad C_Y = BCB^T \quad (18)$$

$Y(t)$ 服从 $N(m_Y, C_Y)$ ，考虑各个路段的交通状况（畅通，堵车或严重堵车）不同，各个路段的行驶时间可以乘以相应的权重，这个权重值可以通过一个1行 n 列的行向量 B 表示。特别的，当 B 是一个1行 n 列的单位行向量（即 $m=1$ ）时， $Y(t)$ 就是所选路线上各个路段行驶时间的总和。那么，按照该路线行走所需要的平均时间为：

$$m_Y = \sum_{i=1}^n m_i \quad (19)$$

一般情况下，根据各个路段的路况信息，可以比较合理的给定通过各个路段的平均时间的权重值，即 $B = [w_{T1} \ w_{T2} \ \dots \ w_{Tn}]$ ，那么可供选择的第 j 条路线需要的平均行驶时间和方差 σ_{Yj}^2 为：

$$m_{Yj} = Bm^T \quad (20)$$

$$\sigma_{Yj}^2 = BCB^T = [w_{T1} \ w_{T2} \ \dots \ w_{Tn}]C[w_{T1} \ w_{T2} \ \dots \ w_{Tn}]^T \quad (21)$$

这时，最优路线的选择按照下式计算：

$$m_{Y0} = \min\{m_{Yj}\} \quad (22)$$

式中： m_{Y0} 表示按照最优路径行驶到达目的地所需要的平均时间， m_{Yj} 表示选择第 j 条路线所需的平均行驶时间。通过计算可以给出保证率为95.44%的平均行驶时间区间 $(m_{Y0} - 2\sigma_{Y0}, m_{Y0} + 2\sigma_{Y0})$ 。

显然，按照这种方法寻优，考虑了各个路段行驶时间的相互影响，但由于需要计算各种可能路线用时并进行比较，其计算效率较低。因此，考虑各个路段相互影响时，该方法适用于节点较少的最优路线选择问题。

四、模型评估

以司机广义行驶费作为最优路径选择的依据，建立了司机广义行驶费用的计算模型。模型为线性方程，考虑了各个路段的平均行驶时间、路段长度以及平均行驶时间的方差等因素，并以权重的形式反映这些因素对广义行驶费用的影响。基于这个模型，根据San Antonio交通网点的分布图，在考虑一组特殊权重值（即线路长度权重 $w_2 = 1$ ）的前提下，用Dijkstra算法编制的程序可以给出任意两个节点间的最优路径。该模型的

特点是物理意义明确，形式简单，便于理解和使用。

五、参考文献

- [1]L.R.Rilett, P.E. and D.Park. Incorporating uncertainty and multiple objectives in real-time route selection [J]. Journal of Transportation Engineering. No.6, 2001.
- [2]盛骤等. 概率论与数理统计[M]. 北京: 高等教育出版社, 1989.
- [3]王光远, 欧进萍. 结构随机振动[M]. 北京: 高等教育出版社, 1998.
- [4]曹立明, 魏兵, 周强. 图论及其在计算机科学中的应用[M]. 徐州: 中国矿业大学出版社, 1995.
- [5]Do H.Nam and Donald R.Drew. Traffic Dynamics: Method for Estimating Freeway Travel Times in Real Time From Flow Measurements [J]. Journal of Transportation Engineering. No.3, 1996.

附录: Optimal Route.exe VB 源程序

VERSION 6.00

Begin VB.Form frmOSPF

```
Caption      = "SELECT THE OPTIMAL ROUTE"  
ClientHeight = 4650  
ClientLeft   = 3360  
ClientTop    = 2910  
ClientWidth  = 7170  
LinkTopic    = "Form2"  
LockControls = -1 'True  
ScaleHeight  = 4650  
ScaleWidth   = 7170
```

Begin VB.TextBox Text1

```
Height      = 3375  
Left        = 240  
MultiLine   = -1 'True  
ScrollBars  = 2 'Vertical  
TabIndex    = 5  
Top         = 1080  
Width       = 6735
```

End

Begin VB.CommandButton Command1

```
Caption      = "PRINT THE OPTIMAL ROUTE"  
Height       = 495  
Left         = 4320  
TabIndex     = 4  
Top         = 360  
Width        = 2055
```

End

Begin VB.ComboBox Combo2

```
Height       = 300  
Left         = 1680  
TabIndex     = 2  
Text         = "DESTINATION"  
Top         = 600  
Width        = 1350
```

End

Begin VB.ComboBox Combo1

```
Height       = 300  
Left         = 90  
TabIndex     = 1  
Text         = "START LOCATION"
```

```

        Top          = 600
        Width       = 1600
    End
    Begin VB.Label Label2
        AutoSize     = -1 'True
        Caption      = "DESTINATION: "
        Height       = 180
        Left         = 1950
        TabIndex     = 3
        Top          = 240
        Width        = 710
    End
    Begin VB.Label Label1
        AutoSize     = -1 'True
        Caption      = "START LOCATION: "
        Height       = 180
        Left         = 240
        TabIndex     = 0
        Top          = 240
        Width        = 720
    End
End
Attribute VB_Name = "frmOSPF"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim start, ends As Long
Dim JLH(1 To 300), LJD(1 To 300, 1 To 4), DST(1 To 300, 1 To 4) As Long
Dim BH(1 To 300), ADD(1 To 300) As String
Dim n As Integer
Dim vertexnum As Integer
Dim edgenum As Integer
Const max = 100000
Dim graph(0 To 300, 0 To 300) As Long
Dim visited(0 To 300) As Integer
Dim path(0 To 300) As Integer
Dim distance(0 To 90000) As Long

Private Function dijkstra(begin As Integer)
    Dim minedge, vertex, i, j, n, m, edges As Integer
    edges = 1
    visited(begin) = 1
    For i = 1 To vertexnum
        distance(i) = graph(begin, i)
        Debug.Print distance(i)
    Next i
    distance(begin) = 0
    While (edges < vertexnum - 1)
        edges = edges + 1
        minedge = max
        For j = 1 To vertexnum
            If visited(j) = 0 And minedge > distance(j) Then
                vertex = j
                minedge = distance(j)
            End If
        Next j
    End While
Next j

```

```

        visited(vertex) = 1
        For n = 1 To vertexnum
            If visited(n) = 0 And (distance(vertex) + graph(vertex, n)) < distance(n) Then
                distance(n) = distance(vertex) + graph(vertex, n)
                path(n) = vertex
            End If
        Next
    Wend
End Function
Private Sub Command1_Click()
    Dim i, j As Integer
    Dim k As Integer
    Dim addname(1 To 100) As String

    Text1 = ""
    If start = 0 Or ends = 0 Then
        MsgBox "PLEASE SELECT START LOCATION AND DESTINATION"
        Exit Sub
    End If
    For i = 1 To vertexnum
        visited(i) = 0
        path(i) = 1
    Next
    dijkstra (start)
    Text1 = "Start location: " & ADD(start) & " --> " & "Destination: " & ADD(ends) & vbCrLf
    Text1 = Text1 & vbCrLf
    If distance(ends) = max Then
        Text1 = Text1 & "There are no avaiable links between them!"
        Exit Sub
    Else
        Text1 = Text1 & "The distance between them is: " & distance(ends) * 0.01 & " mile" & vbCrLf
        Text1 = Text1 & vbCrLf
        Text1 = Text1 & "The optimal route between them is: " & vbCrLf
        Text1 = Text1 & vbCrLf
        Text1 = Text1 & ADD(start)
    End If
    k = ends
    j = 1
    Do
        addname(j) = ADD(k)
        k = path(k)
        j = j + 1
    Loop While (k <> 1)
    For j = j - 1 To 1 Step -1
        Text1 = Text1 & "--> " & addname(j)
    Next j
End Sub

Private Sub Form_Initialize()
    Dim i, j, k As Long
    Dim filename As String
    Dim buffers As String

    On Error Resume Next

    filename = App.path + "\" + "data" + ".txt"
    Open filename For Input As #1
    If LOF(1) = 0 Then

```

```
MsgBox "The lenth of the file is zero! Please Select it asgain。 " & vbCrLf & "The file name is data。 "
```

```
Exit Sub
```

```
End If
```

```
i = 1
```

```
Do While Not EOF(1)
```

```
Line Input #1, buffers
```

```
BH(i) = Val(Mid$(buffers, 1, 3))
```

```
ADD(i) = Mid$(buffers, 4, 4)
```

```
Combo1.AddItem ADD(i)
```

```
Combo2.AddItem ADD(i)
```

```
LJD(i, 1) = Val(Mid$(buffers, 8, 3))
```

```
DST(i, 1) = Val(Mid$(buffers, 11, 3))
```

```
LJD(i, 2) = Val(Mid$(buffers, 14, 3))
```

```
DST(i, 2) = Val(Mid$(buffers, 17, 3))
```

```
LJD(i, 3) = Val(Mid$(buffers, 20, 3))
```

```
DST(i, 3) = Val(Mid$(buffers, 23, 3))
```

```
LJD(i, 4) = Val(Mid$(buffers, 26, 3))
```

```
DST(i, 4) = Val(Mid$(buffers, 29, 3))
```

```
Kill buffers
```

```
i = i + 1
```

```
Loop
```

```
n = i - 1
```

```
Debug.Print n
```

```
Close #1
```

```
vertexnum = n
```

```
edgenum = 0
```

```
For i = 1 To n
```

```
For j = 1 To 4
```

```
If LJD(i, j) = 0 Then Exit For
```

```
edgenum = edgenum + 1
```

```
graph(i, LJD(i, j)) = DST(i, j)
```

```
Next
```

```
Next
```

```
For i = 0 To n
```

```
For j = 0 To n
```

```
If graph(i, j) = 0 Then
```

```
graph(i, j) = max
```

```
End If
```

```
Next
```

```
Next
```

```
End Sub
```

```
Private Sub Combo1_click()
```

```
start = Combo1.ListIndex + 1
```

```
End Sub
```

```
Private Sub Combo2_click()
```

```
ends = Combo2.ListIndex + 1
```

```
End Sub
```