方格路网间两座城市通行的最优道路模型

摘要　　本文针对两座城市之间方格路网通行道路的优化问题进行讨论，建立了二次优化模型和非线性规划模型。利用图与网络分析、非线性规划和概率理论等方法，给出了时间最短路线、费用最小路线以及有时间限制的费用最小路线。并对这几种路线进行比较，分析费用与时间之间的权衡取舍。

首先，通过对 A 城与 B 城之间方格路网的长度、限速、布局的分析，结合运动学定律，在各个道路内部构建时间和费用的一次优化模型。对于每一条道路，优化的结果仅与道路的最高限速有关。对于限速为 50km/h、90 km/h、110 km/h 以及 130 km/h 的道路，最短通行时间分别为 2.00 小时、1.11 小时、0.91 小时以及 0.77 小时，最少通行费用分别为 16.5 元、15.185 元、15.185 元以及 18.185 元。

其次，利用图与网络分析中的贪心算法，构建时间和费用的二次优化模型，进行全局的、整体的优化。将方格路网的 100 个连接点从下至上，从左至右依此标号。则得到的时间最短路线 C1 为 1→2→3→4→14→15→16→26→36→46→56→57→58→59→69→79→89→90→100，共耗时 17.78 小时。得到的花费最小路线 C2 为 1→2→12→22→32→42→52→53→54→55→56→66→76→77→87→88→89→90→100，共需要 274.645 元。

再次，针对存在固定雷达和移动雷达，以及存在严格时间限制的情况，寻找费用最小路线。经过分析，每一条道路的移动雷达个数服从二项分布。利用全概率公式，考虑不同超速条件下罚款的情况，重新定义无固定雷达路段、有固定雷达路段的参数，对上述一次优化模型进行拓展，从而求出每条道路费用的期望。利用上述二次优化模型进行全局优化，得到路线 C3，但是结果并不理想。

然后，以拓展后的一次优化模型为基础，重新建立混合 0-1 变量的非线性规划模型，最终得到耗费时间在 $0.8T$ 内的费用最小路线 C4，为 1→2→12→22→23→33→43→53→54→55→56→66→67→68→69→79→89→90→100，耗费时间恰好为 $0.8T$，也就是 14.224 小时，最小费用为 425.765 元。

最后，本文对路线 C1、C2 以及 C4 进行了比较。路线 C1 比路线 C2 少花了 5.89 个小时，却多花了 18.51 元；路线 C4 比路线 C1 平均每个小时多花了 37.29 元。

此外，本文在最后给出了模型的评价与改进的方向。

关键词　　方格路网；道路优化；贪心算法；非线性规划；概率模型

1. 问题的重述

A 城和 B 城间存在纵横交错的方格路网，横向纵向各有 10 条道路。每一条道路间距离均为 100 公里，但每一段路的限速有所不同。限速在 130 公里/小时的道路为高速路段，收费标准为每段 3 元，而其余路段则无需缴费、免费通行。

除了过路费，旅途中还存在其他的费用。其中一类费用，住店和饮食费用，是时间的函数，这里不妨设$c_1 = 5t$, t为时间的单位（小时）。另一类费用，油费，是汽车行驶速度的函数，不妨以线性函数$c_2 = av + b$进行刻画，v为速度的单位（公里/小时）；此外，油价为 1.3 元/升。

现有两个问题：

（1）若遵守所有的限速，花费时间最短的路线是哪一条？花费路费最少的路线又是哪一条？

（2）交警在某些路段上布置了一些固定雷达，以及 20 个等概率出现在各个路段的移动雷达。每一个雷达都能监控所在的整个路段。当超速 10%时，超速行为被探测到的概率为 0.7，罚款 100 元；当超速 50%时，超速行为被探测到的概率上升为 0.9，此时罚款 200 元。

若因出现紧急情况，需要在第（1）问求得的最短时间的 80%之内赶到 B 城，那么包括罚款在内的花费最少是多少？又应该选择哪条道路？

2. 问题的分析

从 A 城到 B 城的整个方格路网包含 100 个连接点，180 条道路，且各个道路设置了不同的限速，情况繁多、十分复杂，因而难以通过遍历的方法寻求最优路径，只能通过适当的优化方法寻求相应结果。

通过对方格路网的分析，发现每一条道路长度相同，均为 100 公里；而道路上的限速总共有 4 种规格，分别为 50km/h、90km/h、110km/h 以及 130km/h. 想要从 A 城驱车前往 B 城，只能在路网间进行横向和纵向穿行，而不能在没有道路的斜线上穿行。

此外，整个旅途的费用分为“住店和饮食费用”、“油费”以及高速公路过路费。当汽车行驶速度较低时，意味着耗费时间较长，“住店和饮食费用”较高，但与此同时，较低的速度使得“油费”较低；与此相反，当汽车行驶速度提高时，“住店和饮食费用”减少，“油费”则升高。因此，上述两种费用之间存在着平衡关系，而高速公路过路费则仅与道路最高限速有关。

2.1 问题（1）的分析

问题（1）要求求得时间最短路线和费用最小路线。

由于方格路网内的每条道路之间相对独立，汽车在某一条道路上行驶的情况并不会对在另一条道路上的行驶带来影响，因此，对于时间最短的路线，其内部任意一条道路的时间也应该是最短的。基于此种思想，构建一个二次优化模型：

（1）局部优化。首先对方格路网内的 180 条道路分别进行优化。利用运动学定律，在每条道路最高限速的条件下，求得每条道路的最短时间及最优速度；

（2）整体优化。将每条道路的最短时间视作定值，利用贪心算法，求得从 A 城到 B 城的时间最短路线。

对于费用最小路线，利用类似的思想，构建二次优化模型：

（1）局部优化。将“住店和饮食费用”、“油费”以及高速公路过路费进行加总，求出最每条道路的总费用。根据函数关系式求得每条道路的最小费用及最优速度；

（2）整体优化。将每条道路的最小费用视作定值，利用贪心算法，求得从A城到B城的时间最短路线。

在求得两条最优线路之后，可将它们进行比较，对道路的选取细节进行分析和探究。

2.2 问题（2）的分析

问题（2）对路线行驶的时间和花费都提出了要求，需要在时间少和花费少的矛盾中寻求平衡。

由于20个移动雷达在180条道路上出现的概率是相等的，位置的分布是随机的，因此需要引入随机变量对某一条道路上的雷达个数进行刻画，并剔除概率过小的情况，对模型进行简化。之后，利用全概率公式，将问题（1）中的行驶时间、行驶费用模型进行拓展，将随机优化问题转化为期望的优化问题。

在优化过程中，首先考虑仍然使用问题（1）中的二次优化模型，第一步先找出每条道路的最小费用，再利用贪心算法进行全局初探。但是二次优化模型并不能保证求得的路线行驶时间小于$0.8T$。

更加合理的方法，是利用非线性规划的方法寻找费用最小路径。引入0-1变量，一方面将贪心算法转化为非线性规划的表达式，另一方面引入严格的时间约束，将费用函数作为目标函数，使得最终的结果在时间、费用两个方面都满足相应的条件。

最后，将求得的结果进行分析，对道路的选取的细节进行探究和阐释。

3. 模型的假设与符号的说明

3.1 模型的假设

（1）汽车在每一段道路上做匀速直线运动；

（2）汽车在不同道路上行驶的速度可能是不同的；

（3）汽车从一段道路行驶至与另一段道路的连接点处，如果需要改变速度，改变过程是瞬时的，而后立即恢复匀速直线运动；

（4）不计路途中休息时间。

（5）移动雷达在各个道路上的初始分布是随机的，运动速度和方向也是随机的；

（6）移动雷达之间互不影响，相互独立。

3.2 符号的说明

从A城到B城的方格路网中，连接点共有100个，道路共有180条。将这100个点从下至上，从左到右进行编号，如图 1所示。其中，A城编号为1，B城编号为100. 连接点之间的道路则以连接点的编号进行表示。

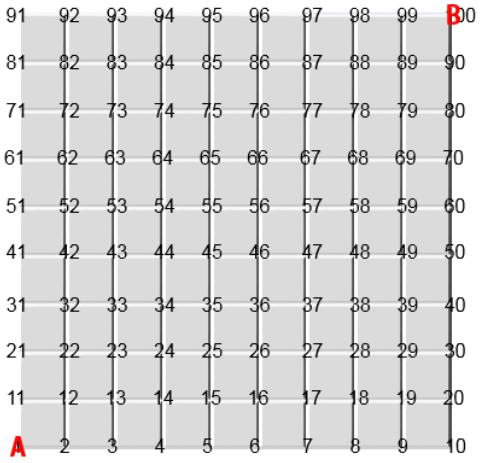

图 1　　方格路网间连接点的编号

表 1　　符号的说明

符号	说明	单位
x_{ij}	从i点到j点之间的道路，为 0-1 变量	—
$v_{ij}^{\max}$	从i点到j点之间道路的最高限速	千米/小时
t_{ij}	汽车在从i点到j点之间道路上行驶的时间	小时
c_{ij}	汽车在从i点到j点之间道路上花费的总费用	元
c_{ij}^1	汽车在从i点到j点之间道路上花费的“住店和饮食”费用	元
c_{ij}^2	汽车在从i点到j点之间道路上花费的“油费”	元
c_{ij}^3	汽车在从i点到j点之间道路上花费的高速公路费用	元
	$i,j=1,2,\ldots,100$，$i\neq j$	

4. 模型的建立、求解与分析

4.1 问题（1）模型的建立、求解与分析

4.1.1 时间最短路线

根据模型的假定，汽车在每一条道路上做匀速直线运动，且换至另一条道路时速度发生改变，改变时间忽略不计。所以可以大致认为，汽车在不同道路之间的行驶情况相对独立。若要使总路线花费时间最短，则必然使得汽车所经过的每一段路花费时间最短。

由运动学定律可知，

$$t_{ij}=\frac{s}{v_{ij}} \tag{1}$$

易得时间与速度成反比。当汽车行驶速度达到最高限速$v_{ij}^{\max}$时，t_{ij}取得最小值$t_{ij}^{\min}$.

注意到，在 A 城到 B 城的方格路网中，所有的道路根据最高限速，可归纳为 4 种规格，分别为 50km/h，90km/h，110km/h 以及 130km/h；则它们的时间最小值如表 2 所示。

表 2　四种限速规格所对应的最小行驶时间

最高限速规格$v_{ij}^{\max}$	$t_{ij}^{\min}$
50 km/h	2.00 h
90 km/h	1.11 h
110 km/h	0.91 h
130 km/h	0.77 h

以$t_{ij}^{\min}$作为从 A 城至 B 城方格路网中，每条道路的权重。

运用图论中的 Dijkstra 算法，得到时间最短路线，如图 2 所示。

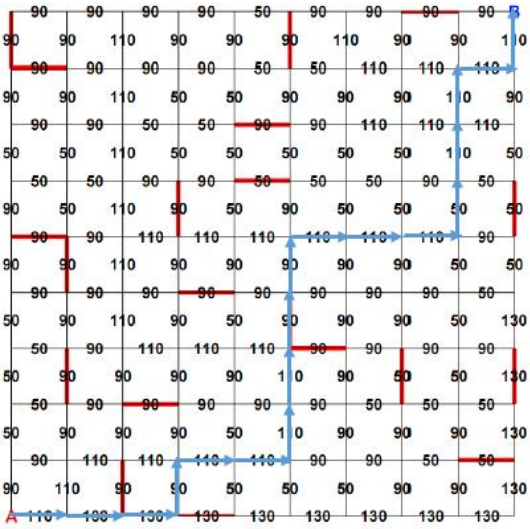

图 2　时间最短路线

时间最短路线为 1→2→3→4→14→15→16→26→36→46→56→57→58→59→69→79→89→90→100，将此路线编号为 C1.

参考表 2，可以看出路线 C1 经过了 1 条限速 50km/h 的道路，3 条限速 90km/h 的道路，12 条限速 110km/h 的道路，以及 2 条限速 130km/h 的道路，总耗时 17.78 小时。

观察从城 A 到城 B 的方格路网，可以大致看出，靠近城 A 的道路（路网偏南部）普遍限速较高，而靠近城 B 的道路（路网偏北部）普遍限速较低。在要求时间最短的情况下，路线 C1 竟然经过了 1 条限速 50km/h 的道路，却仅仅经过 2 条限速 130km/h 的道路；这或许是从南向北的行驶过程中，不能仅仅考虑最高限速，还应考虑行驶的方向，进而进行折中与平衡的结果。

因此，线路 C1 具有较强的合理性和适应性。

4.1.2 花费最少路线

无论是在省道、国道还是高速公路上行驶时，都无可避免地产生各种费用。总的而言，当从某一城市驾车行驶至另一城市时，费用可分为“住店和饮食费用”、“油费”以及高速公路费用。

“住店和饮食费用”指在路途中为避免疲劳驾驶，或者补充能量等，在服务区、客栈等地休整时所产生的花费；路途时间越长，这样的费用也就越高。可较简单地认为，“住店和饮食费用”与时间有如下关系：

$$c_{ij}^1 = 5t_{ij} \tag{2}$$

“油费”指在路途中汽车行驶所消耗的汽油费用。一般而言，汽车行驶速度越高，所耗用的油量也就越多；而当汽车行驶速度为零时，由于汽车内部的空调、电子设备、发动机仍在工作，汽车仍然会耗用一定的油量。可以认为，“油费”与汽车行驶速度存在如下线性关系：

$$c_{ij}^2 = p(av + b) \tag{3}$$

其中p为每升汽油价格，取$p = 1.3$（元/升）；另外，根据题目数据，取$a = 0.0625, b = 1.875.$

此外，从 A 城到 B 城的方格路网中，存在最高限速为 130km/h 的道路。由题目信息所知，这样的道路不同于省道或者国道，它们是收费的高速公路。定义：

$$\delta_{ij} = \begin{cases} 1, & v_{ij}^{\max} = 130 \\ 0, & v_{ij}^{\max} \neq 130 \end{cases}. \tag{4}$$

由于高速公路收费为 3 元/段，故

$$c_{ij}^3 = 3\delta_{ij} \tag{5}$$

将上述三项费用进行加总，可以得到在每一段道路上行驶时，所产生的费用为

$$c_{ij} = c_{ij}^1 + c_{ij}^2 + c_{ij}^3. \tag{6}$$

根据（2）（3）（5）（6）式整理得，

$$\begin{gathered} c_{ij} = \frac{500}{v_{ij}} + 1.3(0.0625v_{ij} + 1.875) + 3\delta_{ij} \\ \text{其中} 0 < v_{ij} \leq v_{ij}^{\max} \end{gathered} \tag{7}$$

针对方格网络中的每一条道路，将所有道路依上述方法、按最高限速分为四种规格，对（7）式求解最小值。

表 3　四种限速规格所对应的最少费用

最高限速规格$v_{ij}^{\max}$	使费用最小的v_{ij}	$c_{ij}^{\min}$
50 km/h	50 km/h	16.5
90 km/h	78.445 km/h	15.185
110 km/h	78.445 km/h	15.185
130 km/h	78.445 km/h	18.185

将每一条道路的最小费用$c_{ij}^{\min}$作为从 A 城至 B 城方格路网中，每条道路的权重。运用 Dijkstra 算法，得到费用最少路线。如图 3 所示。

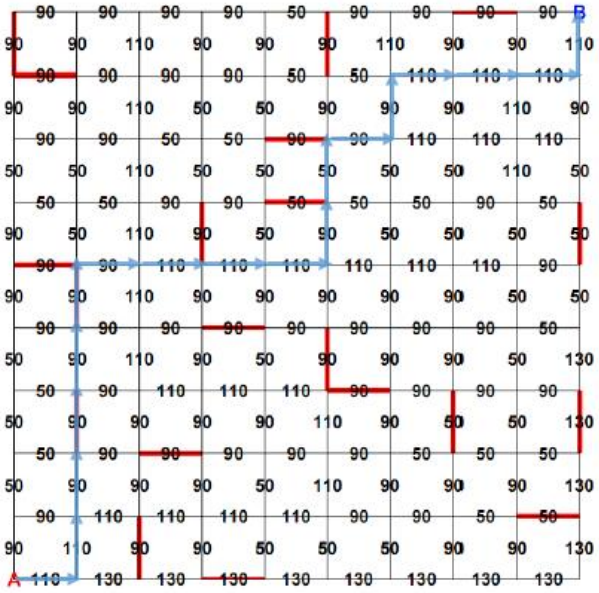

图 3　费用最小路线

费用最小路线为 1→2→12→22→32→42→52→53→54→55→56→66→76→77→87→88→89→90→100，将该线路编号为 C2.

参考表 3，可以看出 C2 经过了 1 条限速 50km/h 的道路，8 条限速 90km/h 的道路，9 条限速 110km/h 的道路，并没有经过限速 130km/h 的道路。路线 C2 总花费为 274.645 元。

在四种规格的道路中，限速 90km/h 的道路和限速 110km/h 的道路费用最低，限速 50km/h 的道路其次，高速公路费用最高。因此，费用最少路线 C2 大多由限速 90km/h 道路和限速 110km/h 道路组成，并没有走高速公路；在由南至北行驶过程中，限速 50km/h 的道路逐渐增多，花费也并非过高，为顺利到达 B 城，选取 1 条该规格的道路也较为合理。

4.1.3 两条路线的对比

将时间最短路线 C1 和费用最少路线 C2 的耗费时间、所花费用进行对比如下：

表 4　　路线 C1 和路线 C2 对比表

编号	路线	耗费时间	所花费用
C1	1→2→3→4→14→15→16→26→36→46→56→57→58→59→69→79→89→90→100	17.78 小时	293.150 元
C2	1→2→12→22→32→42→52→53→54→55→56→66→76→77→87→88→89→90→100	23.67 小时	274.645 元

路线 C1 比路线 C2 少花了 5.89 个小时，却多花了 18.51 元。当出行人认为将近六个小时的时间相对于 18.51 元更加宝贵时，最优的线路显然为 C1。但是对于不赶时间的人，选择路线 C2，能够用时间换金钱，省去 18.51 元，也不失为一个好的选择。

4.2 问题（2）模型的建立、求解与分析

4.2.1 移动雷达的分布

在问题（2）中，为了防止超速行驶，交警在从城 A 到城 B 的方格路网中安排了一些固定雷达和 20 个移动雷达。这 20 个移动雷达等概率地出现在各个路段。在模型的假定下，这 20 个移动雷达在 180 条道路上的初始分布、运动速度和方向是随机的，雷达和雷达之间也互不干扰、相互独立。为了刻画雷达出现的随机性，不妨选定任意一条道路，记随机变量Y_{ij}为该路段移动雷达的个数。

由于：

（1）雷达个数为 20；

（2）对于每个雷达，结果仅有两个，即在该条道路上，或者不在该条道路上；

（3）对于每个雷达，在该条道路上的概率都相等，为$p = \frac{1}{180}$，不在该条道路上的概率也相等，为$1 - p = \frac{179}{180}$；

（4）雷达之间是相互独立的。

所以可以认为，该路段移动雷达的个数服从二项分布，即

$$Y_{ij} \sim b\left(20, \frac{1}{180}\right) \tag{8}$$

计算二项分布下，移动雷达个数的概率分布表：

表 5　　移动雷达个数的概率分布表

k	$P\{Y_{ij} = k\}$
0	0.8946
1	0.09995
2	0.0053
3	0.00018
…	…
20	7.84×10^{-46}

在某一给定道路上，移动雷达个数的概率随着个数增加骤减。所在道路上没有移动雷达是极有可能的，出现 1 个雷达的可能性则骤降至不到 10%。当出现的移动雷达数多于 2 个时，概率已降至约万分之二，基本可以认为这样的小概率事件不会发生。因此，在该模型中，只考虑某一道路上移动雷达个数为 0，1，2 这三种情形。

4.2.2 无固定雷达路段的相关参数

根据题目所给数据，当超速 10%时，超速行为被探测到的概率为 0.7，罚款 100 元；当超速 50%时，超速行为被探测到的概率上升为 0.9，此时罚款 200 元。

对该罚款规则进一步定义，可以明确为：

（1）当$0 < v_{ij} \le 1.1v_{ij}^{\max}$时，不罚款；

（2）当$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$时，罚款 100 元，被罚概率为 70%；

（3）当$v_{ij} > 1.5v_{ij}^{\max}$时，罚款 200 元，被罚概率为 90%.

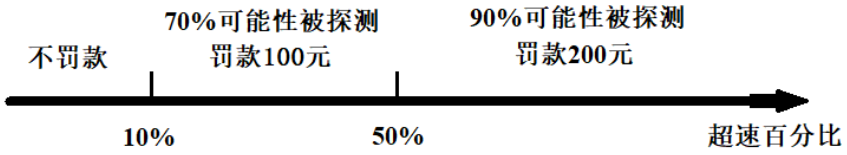

图 4　超速百分比数轴

下面，讨论某一道路移动雷达个数不同时，相应的行驶时间及其花费。

1. 该道路没有移动雷达，即Y_{ij}取$k = 0$.

与问题（1）类似，在该道路上的行驶时间仍可由（1）式表达。

由于道路上不存在移动雷达，当行驶速度超过最高限速$v_{ij}^{\max}$时，并不会被探测、罚款，因此花费表达为

$$c_{ij}^{(0),k=0} = \frac{500}{v_{ij}} + 1.3(0.0625v_{ij} + 1.875) + 3\delta_{ij} \qquad (9)$$

其中行驶速度v_{ij}不受约束。

2. 该道路有一个移动雷达，即Y_{ij}取$k = 1$.

此时，行驶时间仍由（1）式表达；

当行驶速度超过最高限速不到 10%时，由于不存在罚款，花费表达式与（9）式类似；当行驶速度超过最高限速 10%至 50%时，有 70%的可能性被罚款 100 元，求罚款期望，为 70 元；当行驶速度超过最高限速 50%时，有 90%的可能性被罚款 200 元，求罚款期望，为 180 元。

因此，花费可以表达为

$$c_{ij}^{(0),k=1}=\begin{cases}\dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}, & 0<v_{ij}\le 1.1v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.7*100, & 1.1v_{ij}^{\max}<v_{ij}\le 1.5v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.9*200, & v_{ij}>1.5v_{ij}^{\max}\end{cases}\quad（10）$$

3. 该道路有两个移动雷达，即Y_{ij}取$k=2$.

此时，行驶时间仍由（1）式表达；
当行驶速度超过最高限速不到 10%时，由于不存在罚款，花费表达式与（9）式类似。
当行驶速度超过最高限速 10%至 50%时，超速被任意 1 个雷达探测到的可能性为 70%；现在道路上存在 2 个雷达，则至少被 1 个雷达探测到的可能性为$1-(1-0.7)^2=0.91$，罚款的期望为 91 元；
当行驶速度超过最高限速 50%时，超速被任意 1 个雷达探测到的可能性为 90%；现在道路上存在 2 个雷达，则至少被 1 个雷达探测到的可能性为$1-(1-0.9)^2=0.99$，罚款的期望为 198 元。
因此，花费可以表达为

$$c_{ij}^{(0),k=2}=\begin{cases}\dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}, & 0<v_{ij}\le 1.1v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.91*100, & 1.1v_{ij}^{\max}<v_{ij}\le 1.5v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.99*200, & v_{ij}>1.5v_{ij}^{\max}\end{cases}\quad（11）$$

由于上述三种情况究竟哪一种发生是不确定的，故运用全概率公式，求出在没有固定雷达的情况下，每段道路所需费用的期望近似值。

$$E\left(c_{ij}^{(0)}\right)\cong P(Y_{ij}=0)c_{ij}^{(0),k=0}+P(Y_{ij}=1)c_{ij}^{(0),k=1}+P(Y_{ij}=2)c_{ij}^{(0),k=2}\quad（12）$$

4.2.3 有固定雷达路段的相关参数

与无固定雷达路段相比，有固定雷达的路段仅仅多了一个确定的、不随机游走的雷达。固定雷达的存在，使得当汽车超速时，被探测到的概率进一步上升。

1. 该道路没有移动雷达，仅有一个固定雷达，即Y_{ij}取$k=0$.

行驶时间由（1）式表达；
行驶费用为

$$c_{ij}^{(1),k=0}=\begin{cases}\dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}, & 0<v_{ij}\le 1.1v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.7*100, & 1.1v_{ij}^{\max}<v_{ij}\le 1.5v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.9*200, & v_{ij}>1.5v_{ij}^{\max}\end{cases}\quad(13)$$

2. 该道路有一个移动雷达、一个固定雷达，即Y_{ij}取$k=1$.

行驶时间由（1）式表达；
行驶费用为

$$c_{ij}^{(1),k=1}=\begin{cases}\dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}, & 0<v_{ij}\le 1.1v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.91*100, & 1.1v_{ij}^{\max}<v_{ij}\le 1.5v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.99*200, & v_{ij}>1.5v_{ij}^{\max}\end{cases}\quad(14)$$

3. 该道路有两个移动雷达、一个固定雷达，即Y_{ij}取$k=2$.

行驶时间由（1）式表达；
行驶费用为

$$c_{ij}^{(1),k=2}=\begin{cases}\dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}, & 0<v_{ij}\le 1.1v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.973*100, & 1.1v_{ij}^{\max}<v_{ij}\le 1.5v_{ij}^{\max}\\ \dfrac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+0.999*200, & v_{ij}>1.5v_{ij}^{\max}\end{cases}\quad(15)$$

同样地，运用全概率公式，求出在安置了固定雷达的情况下，每段道路所需费用的期望近似值。

$$E\left(c_{ij}^{(1)}\right)\cong P(Y_{ij}=0)c_{ij}^{(1),k=0}+P(Y_{ij}=1)c_{ij}^{(1),k=1}+P(Y_{ij}=2)c_{ij}^{(1),k=2}\quad(16)$$

4.2.4 问题（2）运用 Dijkstra 算法的初探

当整条路线取最小费用时，首先能够想到的，是分别求出每条道路内的最小费用，即对$E\left(c_{ij}^{(0)}\right)$和$E\left(c_{ij}^{(1)}\right)$求关于$v_{ij}$的最小值，再进行进一步优化。

与 4.1.2 中的方法类似，将从 A 城到 B 城的方格路网，按最高限速分为四种规格。此处还需要考虑每一条道路是否安置了固定雷达。每段道路内优化后，所耗费的最小费用和相应的时间如表 6 所示。

表 6　无固定雷达、有固定雷达时，四种限速规格所对应的最少费用

	$v_{ij}^{\max}$	对应最优速度（最少费用）			$c_{ij}^{\min}$	t_{ij}
		$0 < v_{ij} \le 1.1v_{ij}^{\max}$	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	$v_{ij} > 1.5v_{ij}^{\max}$		
无固定雷达	50	55(15.997)	75(22.645)	78.445(34.144)	15.997	1.818
	90	78.445(15.185)	99(22.979)	135(36.094)	15.185	1.275
	110	78.445(15.185)	121(23.848)	165(37.834)	15.185	1.275
	130	78.445(18.185)	143(28.000)	195(42.805)	18.185	1.275
有固定雷达	50	55(15.997)	75(87.431)	78.445(197.062)	15.997	1.818
	90	78.445(15.185)	99(87.765)	135(198.987)	15.185	1.275
	110	78.445(15.185)	121(88.634)	165(200.751)	15.185	1.275
	130	78.445(18.185)	143(92.786)	195(202.722)	18.185	1.275

观察表 6，可以发现：

（1）对于无固定雷达、限速为 50km/h 的道路：由于当汽车超过最高限速不到 10%时，交警不罚款，所以与表 3 相对比，最优速度上升至 55km/h；尽管随着速度的上升（由 55km/h 上升至 78.445km/h），“住店和饮食”和“油费”之间达到平衡、趋于最小，但在“住店和饮食”上节省下来的费用显然无法弥补由于超速而造成的罚款；

（2）对于无固定雷达、限速为 90km/h、110km/h、130km/h 的道路：在不罚款的速度范围内，“住店和饮食”以及“油费”这两项费用已经达到最小，所以提高速度只会徒增罚款；

（3）对于有固定雷达的道路：当车速超过罚款的临界点$1.1v_{ij}^{\max}$时，各个限速规格道路的费用均出现骤增，这是因为固定雷达使得超速行为极易被发现，而罚款至少为其余费用的 7 倍。

综上所述，对每条道路内部进行最小费用优化时，选取的速度依然在罚款的临界点$1.1v_{ij}^{\max}$之下，如表 6 灰底部分所示。

值得注意的是，与表 3 中的情况进行对比，此时对于最高限速为 50km/h 的道路，最优速度上升了 5km/h，这意味着行驶时间将有所下降；与此同时，相对应的最小费用也下降了 0.503 元——这显然是一种更优的情况。基于此，以表 6 中的$c_{ij}^{\min}$作为每条道路的权重，再次运用 Dijkstra 算法，得出费用最小路线，记作 C3.

有趣的是，经过计算，路线 C3 与路线 C2 完全相同。这在一定程度上说明了 Dijkstra 算法的稳定性，即当某种规格道路的行驶速度上升 10%，费用下降 3.048%时，最终结果却没有改变。

表 7　路线 C1、路线 C2 和路线 C3 对比表

编号	路线	耗费时间	所花费用
C1	1→2→3→4→14→15→16→26→36→46→56→57→58→59→69→79→89→90→100	17.78 小时	293.150 元
C2	1→2→12→22→32→42→52→53→54→55→56→66→76→77→87→88→89→90→100	23.67 小时	274.645 元
C3	1→2→12→22→32→42→52→53→54→55→56→66→76→77→87→88→89→90→100	23.49 小时	274.142 元

记$T=17.78$.

遗憾的是，虽然路线 C3 所花费用略有下降，然而耗费的时间非但没有少于$0.8T$，甚至是T的 1.32 倍。显然，路线 C3 并不符合要求。

4.2.5 问题（2）的规划算法

在 4.2.4 中，虽然所得结果路线 C3 的费用最小，但却不满足耗费时间小于$0.8T$的限制，进而使得出行人无法及时从 A 城赶往 B 城。在这种情况下，我们考虑利用规划算法，将出行时间严格控制在$0.8T$之内，在此基础上求得费用最小的路线。

首先，引入 0-1 变量k_0、k_1、k_2，将（12）、（16）分段函数表达式进行简化，使得

$$\begin{aligned} E\left(c_{ij}^{(0)}\right)= & \left(\frac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}\right)P(Y_{ij}=0) \\ & +\left(\frac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+k_1*0.7*100+k_2*0.9*200\right)P(Y_{ij}=1) \\ & +\left(\frac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+k_1*0.91*100+k_1*0.99*200\right)P(Y_{ij}=2) \end{aligned} \tag{17}$$

$$\begin{aligned} E\left(c_{ij}^{(1)}\right)= & \left(\frac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+k_1*0.7*100+k_2*0.9*200\right)P(Y_{ij}=0) \\ & +\left(\frac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+k_1*0.91*100+k_2*0.99*200\right)P(Y_{ij}=1) \\ & +\left(\frac{500}{v_{ij}}+1.3(0.0625v_{ij}+1.875)+3\delta_{ij}+k_1*0.973*100+k_1*0.999*200\right)P(Y_{ij}=2) \end{aligned} \tag{18}$$

对于（17）、（18）式，k_0、k_1、k_2满足

$$\begin{cases} 0\le k_0,k_1,k_2\le 1 \\ k_0+k_1+k_2=1. \\ k_0,k_1,k_2\in Z \end{cases} \tag{19}$$

则v_{ij}的取值范围为

$$1.1v_{ij}^{\max}*k_1+1.5v_{ij}^{\max}*k_2<v_{ij}\le 1.1v_{ij}^{\max}*k_0+1.5v_{ij}^{\max}*k_1+200*k_2 \tag{20}$$

然后，构造优化模型。

以x_{ij}表示从i点到j点之间的道路，为 0-1 变量；当从i点到j点之间的道路不存在，或者i点与j点相邻（即道路存在），但是最终路线不经过这条道路时，令$x_{ij}=0$；当最终路线经过i点与j点的道路时，令$x_{ij}=1$.

以t_{ij}表示从i点到j点之间的时间，若从i点到j点之间的道路不存在，可以认为$t_{ij}=\infty$；以$E(c_{ij})$表示从i点到j点之间的费用的期望，若从i点到j点之间的道路不存在，可以认为$E(c_{ij})=\infty$.

则有

$$\min \sum_{i=1}^{100}\sum_{j=1}^{100} E(c_{ij})x_{ij}$$

$$\text{s.t.}\begin{cases} \sum_{j=1}^{100} x_{1,j}=1 \\ \sum_{i=1}^{100} x_{i,100}=1 \\ \sum_{i=1}^{100} x_{ij}-\sum_{i=1}^{100} x_{ji}=0 \\ \sum_{i=1}^{100}\sum_{j=1}^{100} x_{ij}t_{ij}\le 0.8T \\ x_{ij}\in\{0,1\} \end{cases} \qquad (21)$$

运用 LINGO 编程，解决（17）~（21）式组合而成的规划问题。

从而得到路线 C4：1→2→12→22→23→33→43→53→54→55→56→66→67→68→69→79→89→90→100，如图 5 所示。

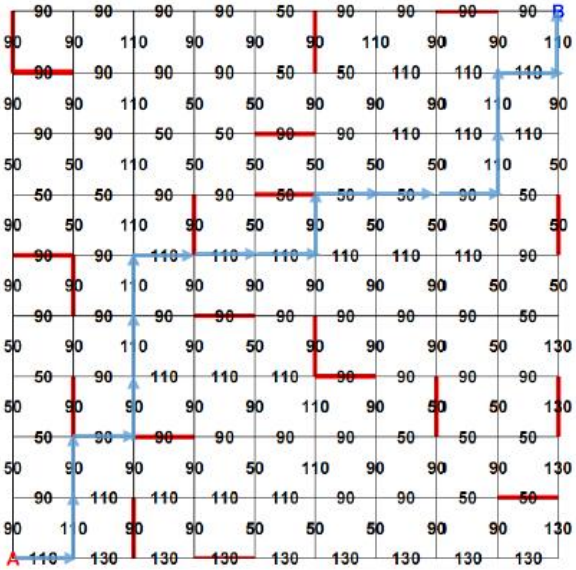

图 5　满足时间少于 0.8T 的费用最少路线

由于采用了规划算法，得到的结果为全局最优解，不再囿于四种最高限速规格的讨论。因此，即使是对于相同规格的道路，汽车行驶的最优速度也会有所不同。路线 C4 的具体情况如下：

表 8　　路线 C4 的具体情况

道路编号	$v_{ij}^{\max}$	有无固定雷达	v_{ij}	超速状态	c_{ij}	t_{ij}
1→2	110	无	118.2095	$0 < v_{ij} \le 1.1v_{ij}^{\max}$	16.272	0.907
2→12	110	无	117.7753	$0 < v_{ij} \le 1.1v_{ij}^{\max}$	16.252	0.821
12→22	90	无	125.2042	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	24.083	0.888
22→23	90	无	121.7537	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	23.915	0.879
23→33	90	无	121.4831	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	23.903	0.876
33→43	110	无	122.1978	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	23.937	0.879
43→53	110	无	131.1115	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	24.383	0.879
53→54	110	无	131.1115	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	24.383	0.626
54→55	110	无	128.3129	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	24.238	0.626
55→56	110	无	128.4322	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	24.245	0.867
56→66	90	无	132.3867	$0 < v_{ij} \le 1.1v_{ij}^{\max}$	16.971	0.867
66→67	50	无	132.3867	$v_{ij} > 1.5v_{ij}^{\max}$	36.011	0.643
67→68	50	无	132.3867	$v_{ij} > 1.5v_{ij}^{\max}$	36.011	0.639
68→69	90	无	170.2246	$v_{ij} > 1.5v_{ij}^{\max}$	38.246	0.639
69→79	110	无	119.2752	$0 < v_{ij} \le 1.1v_{ij}^{\max}$	16.321	0.596
79→89	110	无	119.2752	$0 < v_{ij} \le 1.1v_{ij}^{\max}$	16.321	0.871
89→90	110	无	120.8803	$0 < v_{ij} \le 1.1v_{ij}^{\max}$	16.395	0.871
90→100	110	无	121.0000	$1.1v_{ij}^{\max} < v_{ij} \le 1.5v_{ij}^{\max}$	23.880	0.850
数字特征		均值	127.4115	合计	425.765	14.224

路线 C4 具有如下特征：

（1）路线 C4 中所有道路的行驶速度均超过了最高限速$v_{ij}^{\max}$。这是由于T已经是所有道路达到最高限速时的最短时间，若要求时间小于T，就不得不超速。

（2）路线 C4 完全避开了布置有固定雷达的道路。这是由于固定雷达的存在使得超速行为极易被发现，从而提高被罚款的概率，无法使得费用最小。

（3）路线 C4 中含有 2 条限速 50km/h 的道路、5 条限速 90km/h 的道路以及 11 条限速 110km/h 的道路，但是没有限速为 130 km/h 的道路。这一方面是因为高速路段要另行收费，无法节约费用；另一方面是因为，限速为 90km/h 和 110km/h 道路的罚款临界点，相对于最优的车速几乎足够。

（4）路线 C4 耗费的总时间为 14.224 小时，恰好为 0.8T，比路线 C1 少耗费了 2.556 个小时；这说明时间的约束是一个紧约束。

（5）路线 C4 的总费用为 425.765 元，比路线 C1 的总费用多了 132.615 元。可以认为，这 132.615 元是需要为行驶时间缩短 20%而付出的代价。

事实上，路线 C4 相对于路线 C1，平均每个小时多花了 37.29 元。出行人从 A 城赶往 B 城处理的紧急事件回报率如何，是否达到了 37.29 元/小时？如果没有达到，出行人损失的或许不仅仅是超速罚款，超速所带来的安全隐患、违章记录也同样值得思量。当然，如果事关重大，回报率极高，选择路线 C4 及相应的

速度进行行驶，也不失为一种好的选择。

5. 模型的评价与改进方向

5.1 模型的优点

（1）利用简单的运动学定律和图论中的 Dijkstra 算法，建立二次优化模型，先局部优化再整体优化，得到了较为合理的结果；

（2）利用二项分布对移动雷达的随机位置进行刻画，并运用全概率公式，有效地反映了汽车在路网中行驶的时间、费用情况；

（3）以两种不同的角度——二次优化模型以及规划模型，思考在严格时间限制的条件下取得最小费用线路的问题，提供了不同的视角，对问题的考虑较为全面；

（4）对求解得到的各条路线进行进一步分析，为出行人进行决策提供了更为全面的依据。

5.2 模型的缺点及改进方向

（1）模型的假设较为粗糙。事实上，汽车在路网中行驶时很难全程保持匀速直线运动，从一条道路行驶至另一道路时，速度的改变也不会瞬时完成；此外，在将近 20 个小时的行程中，驾驶员总会需要停车休息。因此，将道路连接点之间的休息、变速时间纳入考虑是模型改进的方向；

（2）最优线路可能存在多条。在从 A 城至 B 城的方格路网中，各个道路长度一致，而限速规格仅有四种，存在极大的相似性。因此，问题（1）中的时间最短路线以及费用最小路线应存在多条。对这多条最优线路进行归纳是模型改进的方向；

（3）忽略了汽车本身最高行驶速度的限制。问题（2）的规划结果中，从连接点 68 到 69 的最优速度达到了惊人的 170.2246km/h；汽车能否达到这样速度，以及路况是否允许这样高速行驶值得商榷；

（4）缺乏对某些参数的灵敏度分析。例如，当交通规则出现变化，对不同超速百分比所对应的罚款数额进行更改时，最优路线的变化问题，是模型需要进一步讨论的方向。

参考文献

[1] 姜启源，谢金星，叶俊. 数学模型（第四版）. 北京：高等教育出版社，2011.
[2] 胡运权，郭辉煌. 运筹学教程（第四版）. 北京：清华大学出版社，2012.
[3] 司守奎，孙玺菁. 数学建模算法与应用. 北京：国防工业出版社，2011.
[4] 袁新生，邵大宏，郁时炼. LINGO 和 Excel 在数学建模中的应用. 北京：科学出版社，2007.

附 录

Matlab 程序代码

a1_1

```
clc, clear
a=zeros(100,100);
a(1,2)=110;a(2,3)=130;a(3,4)=130;a(4,5)=130;a(5,6)=130;a(6,7)=130;a(7,8)=130
;a(8,9)=130;a(9,10)=130;
a(11,12)=90;a(12,13)=110;a(13,14)=110;a(14,15)=110;a(15,16)=110;a(16,17)=9
0;a(17,18)=90;a(18,19)=50;a(19,20)=50;
a(21,22)=50;a(22,23)=90;a(23,24)=90;a(24,25)=90;a(25,26)=90;a(26,27)=90;a(2
7,28)=50;a(28,29)=50;a(29,30)=50;
a(31,32)=50;a(32,33)=90;a(33,34)=110;a(34,35)=110;a(35,36)=110;a(36,37)=90;
a(37,38)=90;a(38,39)=90;a(39,40)=90;
a(41,42)=90;a(42,43)=90;a(43,44)=90;a(44,45)=90;a(45,46)=90;a(46,47)=90;a(4
7,48)=90;a(48,49)=90;a(49,50)=50;
a(51,52)=90;a(52,53)=90;a(53,54)=110;a(54,55)=110;a(55,56)=110;a(56,57)=11
0;a(57,58)=110;a(58,59)=110;a(59,60)=90;
a(61,62)=50;a(62,63)=50;a(63,64)=90;a(64,65)=90;a(65,66)=50;a(66,67)=50;a(6
7,68)=50;a(68,69)=90;a(69,70)=50;
a(71,72)=90;a(72,73)=90;a(73,74)=50;a(74,75)=50;a(75,76)=90;a(76,77)=90;a(7
7,78)=110;a(78,79)=110;a(79,80)=110;
a(81,82)=90;a(82,83)=90;a(83,84)=90;a(84,85)=90;a(85,86)=50;a(86,87)=50;a(8
7,88)=110;a(88,89)=110;a(89,90)=110;
a(91,92)=90;a(92,93)=90;a(93,94)=90;a(94,95)=90;a(95,96)=50;a(96,97)=90;a(9
7,98)=90;a(98,99)=90;a(99,100)=90;

a(1,11)=90;a(2,12)=110;a(3,13)=90;a(4,14)=90;a(5,15)=50;a(6,16)=50;a(7,17)=5
0;a(8,18)=90;a(9,19)=90;a(10,20)=130;
a(11,21)=50;a(12,22)=90;a(13,23)=90;a(14,24)=90;a(15,25)=50;a(16,26)=110;a(
17,27)=90;a(18,28)=90;a(19,29)=90;a(20,30)=130;
a(21,31)=50;a(22,32)=90;a(23,33)=90;a(24,34)=90;a(25,35)=90;a(26,36)=110;a(
27,37)=90;a(28,38)=50;a(29,39)=50;a(30,40)=130;
a(31,41)=50;a(32,42)=90;a(33,43)=110;a(34,44)=90;a(35,45)=50;a(36,46)=90;a(
37,47)=50;a(38,48)=90;a(39,49)=50;a(40,50)=130;
a(41,51)=90;a(42,52)=90;a(43,53)=110;a(44,54)=90;a(45,55)=90;a(46,56)=90;a(
47,57)=90;a(48,58)=90;a(49,59)=50;a(50,60)=50;
a(51,61)=90;a(52,62)=50;a(53,63)=110;a(54,64)=90;a(55,65)=50;a(56,66)=90;a(
57,67)=50;a(58,68)=50;a(59,69)=50;a(60,70)=50;
a(61,71)=50;a(62,72)=50;a(63,73)=110;a(64,74)=50;a(65,75)=50;a(66,76)=50;a(
67,77)=50;a(68,78)=50;a(69,79)=110;a(70,80)=50;
a(71,81)=90;a(72,82)=90;a(73,83)=110;a(74,84)=50;a(75,85)=50;a(76,86)=90;a(
77,87)=90;a(78,88)=90;a(79,89)=110;a(80,90)=90;
```

a(81,91)=90;a(82,92)=90;a(83,93)=110;a(84,94)=90;a(85,95)=90;a(86,96)=90;a(87,97)=110;a(88,98)=90;a(89,99)=90;a(90,100)=110;

a=a' ; %matlab 工具箱要求数据是下三角矩阵

b=100./a;
[i,j,v]=find(b);
c=sparse(a)%构造稀疏矩阵，只储存非零元素及其位置，节约储存空间。用b=full(b)可转化为满阵
[x,y,z]=graphshortestpath(c,1,100,'Directed',false) % Directed 是标志图为有向或无向的属性，该图是无向图，对应的属性值为 false，或 0。
h=view(biograph(c,[],'ShowArrows','off','ShowWeights','on'))

a1_2

```
% clear%求非线性多元函数无约束极值
% ff=@(x)sin(x(1))+cos(x(2));
% x0=[0,0];
% [sx,sfval,sexit,soutput]=fminsearch(ff,x0)
clear
clc
f=@(v)500./v+(0.0625*v+1.875)*1.3;
[v,fimn]=fminbnd(f,0,150)
ezplot(f,[0,150])
```

a1_3

```
clc,clear
a=zeros(100,100);
a(1,2)=110;a(2,3)=130;a(3,4)=130;a(4,5)=130;a(5,6)=130;a(6,7)=130;a(7,8)=130;a(8,9)=130;a(9,10)=130;
a(11,12)=90;a(12,13)=110;a(13,14)=110;a(14,15)=110;a(15,16)=110;a(16,17)=90;a(17,18)=90;a(18,19)=50;a(19,20)=50;
a(21,22)=50;a(22,23)=90;a(23,24)=90;a(24,25)=90;a(25,26)=90;a(26,27)=90;a(27,28)=50;a(28,29)=50;a(29,30)=50;
a(31,32)=50;a(32,33)=90;a(33,34)=110;a(34,35)=110;a(35,36)=110;a(36,37)=90;a(37,38)=90;a(38,39)=90;a(39,40)=90;
a(41,42)=90;a(42,43)=90;a(43,44)=90;a(44,45)=90;a(45,46)=90;a(46,47)=90;a(47,48)=90;a(48,49)=90;a(49,50)=50;
a(51,52)=90;a(52,53)=90;a(53,54)=110;a(54,55)=110;a(55,56)=110;a(56,57)=110;a(57,58)=110;a(58,59)=110;a(59,60)=90;
a(61,62)=50;a(62,63)=50;a(63,64)=90;a(64,65)=90;a(65,66)=50;a(66,67)=50;a(67,68)=50;a(68,69)=90;a(69,70)=50;
a(71,72)=90;a(72,73)=90;a(73,74)=50;a(74,75)=50;a(75,76)=90;a(76,77)=90;a(77,78)=110;a(78,79)=110;a(79,80)=110;
a(81,82)=90;a(82,83)=90;a(83,84)=90;a(84,85)=90;a(85,86)=50;a(86,87)=50;a(8
```

```
7,88)=110;a(88,89)=110;a(89,90)=110;
    a(91,92)=90;a(92,93)=90;a(93,94)=90;a(94,95)=90;a(95,96)=50;a(96,97)=90;a(9
7,98)=90;a(98,99)=90;a(99,100)=90;

    a(1,11)=90;a(2,12)=110;a(3,13)=90;a(4,14)=90;a(5,15)=50;a(6,16)=50;a(7,17)=5
0;a(8,18)=90;a(9,19)=90;a(10,20)=130;
    a(11,21)=50;a(12,22)=90;a(13,23)=90;a(14,24)=90;a(15,25)=50;a(16,26)=110;a(
17,27)=90;a(18,28)=90;a(19,29)=90;a(20,30)=130;
    a(21,31)=50;a(22,32)=90;a(23,33)=90;a(24,34)=90;a(25,35)=90;a(26,36)=110;a(
27,37)=90;a(28,38)=50;a(29,39)=50;a(30,40)=130;
    a(31,41)=50;a(32,42)=90;a(33,43)=110;a(34,44)=90;a(35,45)=50;a(36,46)=90;a(
37,47)=50;a(38,48)=90;a(39,49)=50;a(40,50)=130;
    a(41,51)=90;a(42,52)=90;a(43,53)=110;a(44,54)=90;a(45,55)=90;a(46,56)=90;a(
47,57)=90;a(48,58)=90;a(49,59)=50;a(50,60)=50;
    a(51,61)=90;a(52,62)=50;a(53,63)=110;a(54,64)=90;a(55,65)=50;a(56,66)=90;a(
57,67)=50;a(58,68)=50;a(59,69)=50;a(60,70)=50;
    a(61,71)=50;a(62,72)=50;a(63,73)=110;a(64,74)=50;a(65,75)=50;a(66,76)=50;a(
67,77)=50;a(68,78)=50;a(69,79)=110;a(70,80)=50;
    a(71,81)=90;a(72,82)=90;a(73,83)=110;a(74,84)=50;a(75,85)=50;a(76,86)=90;a(
77,87)=90;a(78,88)=90;a(79,89)=110;a(80,90)=90;
    a(81,91)=90;a(82,92)=90;a(83,93)=110;a(84,94)=90;a(85,95)=90;a(86,96)=90;a(
87,97)=110;a(88,98)=90;a(89,99)=90;a(90,100)=110;
    a=a';
    for i=1:100
        for k=1:100
            if(a(i,k)==50)
                a(i,k)=16.5
            elseif(a(i,k)==90||a(i,k)==110)
                a(i,k)=15.185;
            elseif(a(i,k)==130)
                a(i,k)=18.185;
            end
        end
    end
    [i,j,v]=find(a);
    c=sparse(i,j,v,100,100);%构造稀疏矩阵，只储存非零元素及其位置，节约储存
空间。用 b=full(b)可转化为满阵
    [x,y,z]=graphshortestpath(c,1,100,'Directed',false) % Directed 是标志图为有向
或无向的属性，该图是无向图，对应的属性值为 false，或 0。
    h = view(biograph(c,[],'ShowArrows','off','ShowWeights','on'));
    % Edges = getedgesbynodeid(h); %提取句柄 h 中的边集
    % set(Edges,'LineColor',[0 0 0]); %为了将来打印清楚，边画成黑色
    % set(Edges,'LineWidth',1.5);   %线型宽度设置为 1.5
```

LINGO 程序代码

```
!<pre name="code" class="plain">model:;
model:

sets:
node/1..100/:no;

! V:速度

Vmax: 最大速度

X: 是否选择,

C: 是高速公路为3，否则为0

G: 为1时有固定雷达，为0时无;

link(node,node):V,Vmax,C,X,G;

endsets

data:

Vmax=@ole('D:\Matlab\bin\vmax.xls','vmax');

!高速公路为3，其他路为0，无路为无穷大;

C =@ole('D:\Matlab\bin\vmax.xls','ci');

G =@ole('D:\Matlab\bin\vmax.xls','G')

enddata

! 对每个约束条件的定义;

   min=@sum(node(i):@sum(node(j): @if(V(i,j)#le#1.1*Vmax(i,j),C +
5*100/V(i,j) + 1.3*(0.0625*V(i,j) + 1.875)
   ,@if(V(i,j)#le#1.5*Vmax(i,j),C + 5*100/V(i,j) +
1.3*(0.0625*V(i,j) + 1.875) + 100 * ((0.09995+G(i,j)*0.79465)*0.7 +
(0.0053+G(i,j)*0.09465)*0.91 + 0.0053*G(i,j)*0.973 )
   ,C + 5*100/V(i,j) + 1.3*(0.0625*V(i,j) + 1.875) + 200 *
((0.09995+G(i,j)*0.79465)*0.9 + (0.0053+G(i,j)*0.09465)*0.99 +
0.0053*G(i,j)*0.999 ))   )*X(i,j))  ) ;

   !(0.8946*0.7 + 0.09995*0.91 + 0.0053*0.973);
   !0.8946 - 0.09995 = 0.79465,  0.09995- 0.0053 = 0.09465;
```

```
    !((0.8946+G(i,j)*0.1054)*0.7 + (0.09995++G(i,j)*0.79465)*0.91 +
(0.0053+0.09465*G(i,j))*0.973 + 0.0053*G(i,j)*0.9919 );
    !200 * (0.8946*0.9 + 0.09995*0.99 + 0.0053*0.999) ;
    !200 * ((0.8946+G(i,j)*0.1054)*0.9 +
(0.09995++G(i,j)*0.79465)*0.99 + (0.0053+0.09465*G(i,j))*0.999 +
0.0053*G(i,j)*0.9999 );

    X(1,2) + X(1,11) =1;
    X(2,1) + X(11,1) =0;
    !@sum(node(i): X(i,100)) =1;
    X(99,100) + X(90,100) = 1;
    X(100,99) + x(100,90) = 0;
    @for(node(j)|j#ge#2 #and# j#le#99: @sum(node(i): X(i,j))   =
@sum(node(k): X(j,k)) ) ;
    @sum(node(i): @sum(node(j): X(i,j) * 100/V(i,j) )  ) <= 0.8*17.78;

    @for(node(i): @for(node(j):
V(i,j)<=@if(Vmax(i,j)#le#50,100,200) ));
    @for(link:@bin(X));
    end
```