Appendix D: Programming Sources for the ITPM

D.1 C^{++} Programming Code for Lowry Model.

```
//******************************************************************************
// program : Lowry Model
// Discription of program
//******************************************************************************
#include <iostream.h>
#include <fstream.h>   //for file I/O
#include <iomanip.h>
int main()
{
   int i, j, nZone, itration ;
           float BE[12], SE[12], TE[12], TP[12], TT[12][12], probint0[12][12], probint1[12][12],
                       tprobint[12], BEZ[12][12],TBEZ[12], PN[12], DN[12], GPN[12], S[12][12], TS[12],
                       GTS[12], GTB[12], GTGTS, GTGPN, GTGTB, TBE, TSE, TTE, TTP, alpha, beta;
           ifstream INPUT1, INPUT2;
           ofstream OUTPUT;

           INPUT1.open("Emppop12.dat");
   INPUT2.open("Tt12.dat");
   OUTPUT.open("Lowry4.txt");
           nZone=11;
//data input
   itration=1;
   TBE=0.;
   TSE=0.;
   TTE=0.;
   TTP=0.;
   alpha=0.;
   beta=0.;
           i=1;
           do{
           GPN[i]=0.;
           BE[i]=0.;
           SE[i]=0.;
           TE[i]=0.;
           TP[i]=0.;
   INPUT1 >>BE[i] >>SE[i] >>TE[i] >>TP[i];
   TBE=TBE+BE[i];
   TSE=TSE+SE[i];
   TTE=TTE+TE[i];
   TTP=TTP+TP[i];
i=i+1;
           }while(i<=nZone);
           i=1;
           do{
           TT[i][1]=0.;
           TT[i][2]=0.;
           TT[i][3]=0.;
           TT[i][4]=0.;
           TT[i][5]=0.;
           TT[i][6]=0.;
           TT[i][7]=0.;
           TT[i][8]=0.;
           TT[i][9]=0.;
           TT[i][10]=0.;
           TT[i][11]=0.;
   INPUT2
>>TT[i][1]>>TT[i][2]>>TT[i][3]>>TT[i][4]>>TT[i][5]>>TT[i][6]>>TT[i][7]>>TT[i][8]>>TT[i][9]>>TT[i][10]>>TT[i][11];
i=i+1;
```

```
            }while(i<=nZone);
             i=1;
            alpha=TTP/TTE;
            beta =TSE/TTP;
            do{
   tprobint[i]=0.;
            j=1;
            do{
            probint0[i][j]=TP[j]/(TT[i][j]*TT[i][j]);
            tprobint[i]=tprobint[i]+probint0[i][j];
            j=j+1;
            }while(j<=nZone);
            i=i+1;
            }while(i<=nZone);
            i=1;
            do{
            j=1;
            do{
            probint0[i][j]=probint0[i][j]/tprobint[i];
            j=j+1;
            }while(j<=nZone);
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=1;
            do{
            TBEZ[j]=0.;
            do{
            BEZ[i][j]=BE[i]*probint0[i][j];
            TBEZ[j]=TBEZ[j]+BEZ[i][j];
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=j+1;
            }while(j<=nZone);
            i=1;
            j=1;
//Basic Population PN & Service employ demand by the pop. DN//
            do{
            PN[i]=alpha*TBEZ[i];
            GPN[i]=GPN[i]+PN[i];
            DN[i]=beta*PN[i];
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=1;
            do{
   tprobint[j]=0.;
            do{
            probint1[i][j]=SE[i]/(TT[i][j]*TT[i][j]);
            tprobint[j]=tprobint[j]+probint1[i][j];
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=j+1;
            }while(j<=nZone);
            i=1;
            j=1;
            do{
            j=1;
            do{
            probint1[i][j]=probint1[i][j]/tprobint[j];
            j=j+1;
            }while(j<=nZone);
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=1;
            do{
            j=1;
            TS[i]=0.;
            GTS[i]=0.;
            do{
            S[i][j]=DN[j]*probint1[i][j];
            TS[i]=TS[i]+S[i][j];
            j=j+1;
            }while(j<=nZone);
   GTS[i]=GTS[i]+TS[i];
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=1;
// ITERATIONS //
            do{
            do{
            TBEZ[j]=0.;
            do{
            BEZ[i][j]=TS[i]*probint0[i][j];
            TBEZ[j]=TBEZ[j]+BEZ[i][j];
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=j+1;
            }while(j<=nZone);
            i=1;
            j=1;
//Basic Population PN & Service employ demand by the pop. DN//
            do{
            PN[i]=alpha*TBEZ[i];
            GPN[i]=GPN[i]+PN[i];
            DN[i]=beta*PN[i];
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=1;
            do{
            j=1;
            TS[i]=0;
            do{
            S[i][j]=DN[j]*probint1[i][j];
            TS[i]=TS[i]+S[i][j];
            j=j+1;
            }while(j<=nZone);
   GTS[i]=GTS[i]+TS[i];
            i=i+1;
            }while(i<=nZone);
            i=1;
            j=1;
            itration=itration+1;
   }while(itration<=16);
   GTGTS=0.;
   GTGTB=0.;
   GTGPN=0.;
            OUTPUT <<"           SERVICE EMP.      TOTAL EMP.      TOTAL POP."<<endl;
            OUTPUT <<"--------------------------------------------------------"<<endl;
            OUTPUT <<" ZONE        act. predict.   act. predict.   act. predict."<<endl;
            OUTPUT <<"--------------------------------------------------------"<<endl;
            do{
            GTB[i]=BE[i]+GTS[i];
            GTGTS=GTGTS+GTS[i];
            GTGTB=GTGTB+GTB[i];
            GTGPN=GTGPN+GPN[i];
            OUTPUT <<" ["<<setw(2)<<i<<"]
"<<setiosflags(ios::fixed)<<setw(9)<<setprecision(0)<<SE[i]<<setw(8)<<GTS[i]<<setw(8)<<TE[i]<<setw(8)<<GTB[i]<<setw(8)
<<TP[i]<<setw(8)<<GPN[i]<<endl;
            i=i+1;
            }while(i<=11);
            OUTPUT <<"--------------------------------------------------------"<<endl;
            OUTPUT <<"TOTAL
"<<setiosflags(ios::fixed)<<setw(9)<<setprecision(0)<<TSE<<setw(8)<<GTGTS<<setw(8)<<TTE<<setw(8)<<GTGTB<<setw(8
)<<TTP<<setw(8)<<GTGPN<<endl;

return 0;
}
```

Table D.1 Input Data for the Lowry Model (Zonal Employment; Persons).

Zone	Basic Employment	Service Employment	Total Employment	Population
1	106	27	133	842
2	197	51	248	1574
3	1281	256	1537	3145
4	763	197	961	4027
5	6315	1633	7948	10409
6	543	291	834	3631
7	889	475	1364	2626
8	1230	318	1548	4059
9	362	93	455	901
10	292	75	367	5529
11	14	3	17	912

Table D.2 Input Data for the Lowry Model (Interzonal Travel Time; Min.).

Zone	1	2	3	4	5	6	7	8	9	10	11
1	0.5	12	5.9	6.5	9.1	10.1	14	7.9	17.9	7.4	13.2
2	12	2.5	6.3	6.6	3.8	2.4	2.2	5.8	7.8	6.7	6.8
3	5.9	6.3	2.5	3.2	4.4	4.2	8.2	3.2	13.4	8.4	6.6
4	6.5	6.6	3.2	6.5	3.9	4.3	8.6	4.3	13.1	7.5	3.7
5	11.8	4.9	5.7	5.1	4	4.2	7.8	4.2	12.1	5.6	10.2
6	10.1	2.4	4.2	4.3	3.2	1.5	4.1	4.8	10.4	7.6	5
7	14	2.2	8.2	8.6	6	4.1	1.5	7.9	8	6.4	6.2
8	7.9	5.8	3.2	4.3	4.2	4.8	7.9	13	11.1	5.2	5.4
9	17.9	7.8	13.4	13.1	9.3	10.4	8	11.1	1.5	5.1	14.2
10	7.4	6.7	8.4	7.5	5.6	7.6	6.4	5.2	5.1	0.5	7.3
11	13.2	6.8	6.6	3.7	10.2	5	6.2	5.4	11.2	7.3	0.9

Table D.3 C^{++} Lowry Model Program Output (Persons).

ZONE	SERVICE EMP.		TOTAL EMP.		TOTAL POP.	
	act.	predict.	act.	predict.	act.	predict.
[1]	27	40	133	146	842	489
[2]	51	57	248	254	1574	1677
[3]	256	295	1537	1576	3145	3264
[4]	197	173	961	936	4027	3312
[5]	1633	1277	7948	7592	10409	11865
[6]	291	498	834	1041	3631	4854
[7]	475	418	1364	1307	2626	2571
[8]	318	248	1548	1478	4059	3905
[9]	93	47	455	409	901	570
[10]	75	362	367	654	5529	4763
[11]	3	3	17	17	912	381
TOTAL	3419	3419	15412	15411	37655	37652

D.2 C^{++} Programming Code for Trip Generation Model.

```
//****************************************************************
// program : TRIP GENERATION/DISTRIBUTION
// Discription of program
//****************************************************************
#include <iostream.h>
#include <fstream.h>   //for file I/O
int main()
{
        int i, j, n, nZone, itration, k, m, buffer ;
        double O[12], D[12], SUMJOB, SUMAH, A[12], B[12], AH[12],  cap[12],
           tt[12][12], Bdash[12], Adash[12], Vdash[12][12], forAdash,
           forBdash, Osum[12], Dsum[12];
                                                     //O[0,..93]=  L[0,..93]

        //D[0,..93]=job[0,..93]
        ifstream INPUT1, INPUT2;
        ofstream OUTPUT;
         INPUT1.open("od11.dat");
  INPUT2.open("tt11.dat");
  OUTPUT.open("for11.txt");
        i=0;
        nZone=11;
        SUMJOB=0.;
        SUMAH=0.;
        forAdash=0.;
        forBdash=0.;
 do{
        AH[i]=0.;
        A[i]=0.;
  O[i]=0.;
  D[i]=0.;
  A[i]=0.;
  B[i]=0.;
  AH[i]=0.;
  D[i]=0.;
  cap[i]=0.;
  Bdash[i]=0.;
  Adash[i]=0.;
  Osum[n]=0;
  Dsum[n]=0;
        j=0;
 do{
  tt[i][j]=0.;
  Vdash[i][j]=0.;
  j=j+1;
  }while(j<=nZone);
  i=i+1;
        }while(i<=nZone);
        i=1;
        do{
  INPUT1 >>O[i];
  INPUT1 >>D[i];
  SUMJOB=SUMJOB+D[i];
  cout<<"O["<<i<<"]= "<<O[i]<<" D["<<i<<"]= "<<D[i]<<endl;
  i=i+1;
        }while(i<=nZone);
        i=1;
        do{
  INPUT2 >>tt[i][1] >>tt[i][2] >>tt[i][3] >>tt[i][4] >>tt[i][5] >>tt[i][6]
      >>tt[i][7] >>tt[i][8] >>tt[i][9] >>tt[i][10]>>tt[i][11];
        i=i+1;
        }while(i<=nZone);

        i=1;
        do{
        A[i]=0.;
        j=1;
        do{
        if (tt[i][j]!=0.)
        {
  A[i]=A[i]+D[j]/tt[i][j];
  j=j+1;
  }
        else
        {
                j=j+1;
        }
        }while(j<=nZone);
  i=i+1;
        }while(i<=nZone);
  SUMAH=0.;
        i=1;
        do{
  AH[i]=A[i]*cap[i];
  SUMAH=SUMAH+AH[i];
  i=i+1;
        }while(i<=nZone);

        i=1;
        do{
//      O[i]=0.;
//   O[i]=AH[i]*SUMJOB/SUMAH;
  OUTPUT <<"O["<<i<<"]"<<"   "<<O[i]<<"\n";
  cout<<"O,,["<<i<<"]"<<"   "<<O[i]<<endl;
  i=i+1;
        }while(i<=nZone);
//*****************************************************************
// program : TRIP DISTRIBUTION
// Discription of program
//*****************************************************************
        itration=1;
        do
        {
        i=1;
         OUTPUT <<"itration =  "<<itration<<"\n";
         do
         {
          if ((D[i]!=0.)&&(itration==1))
          {
           Bdash[i]=1.;
          }
          else
          {
           if(D[i]!=0.)
           {
            j=1;
            do
```

```
              {
               if(tt[j][i]!=0.)
               {
                forBdash=forBdash+Adash[j]*O[j]/tt[j][i];
               }
               else
               {
                buffer=buffer;
               }
                j=j+1;
              }while(j<=nZone);
              if(forBdash!=0.)
              {
               Bdash[i]=1/forBdash;
               forBdash=0.;
              }
              else
              {
               buffer=buffer;
  }
             }
             else
             {
  buffer=buffer;
 }
}
i=i+1;
}while(i<=nZone);

i=1;
do
{
            if ((O[i]!=0.)&&(itration==1))
            {
             Adash[i]=1/A[i];
}
            else
            {
             if(O[i]!=0.)
             {
              j=1;
              do
              {
               if(tt[i][j]!=0.)
               {
                forAdash=forAdash+Bdash[j]*D[j]/tt[i][j];
               }
               else
               {
                buffer=buffer;
               }
                j=j+1;
              }while(j<=nZone);
              if(forAdash!=0.)
              {
               Adash[i]=1/forAdash;
               forAdash=0.;
              }
              else
              {
                    buffer=buffer;
        }
                   }
                   else
                   {
        buffer=buffer;
        }
       }
                 i=i+1;
                }while(i<=nZone);
                k=1;
                do
                {
                 m=1;
       do
                 {
                  if(tt[k][m]!=0.)
                  {
                   Vdash[k][m]=Adash[k]*Bdash[m]*O[k]*D[m]/tt[k][m];
                  }
                  else
                  {
                   buffer=buffer;
                  }
                  m=m+1;
       }while(m<=nZone);
       k=k+1;
      }while(k<=nZone);
                OUTPUT <<"     1     2     3     4     5     6     7     8     9    10    11    Oi"<<"\n";
      n=1;
      do
      {
       Osum[n]=Vdash[n][1]+Vdash[n][2]+Vdash[n][3]+Vdash[n][4]
             +Vdash[n][5]+Vdash[n][6]+Vdash[n][7]+Vdash[n][8]
             +Vdash[n][9]+Vdash[n][10]+Vdash[n][11];
       Dsum[n]=Vdash[1][n]+Vdash[2][n]+Vdash[3][n]+Vdash[4][n]
             +Vdash[5][n]+Vdash[6][n]+Vdash[7][n]+Vdash[8][n]
             +Vdash[9][n]+Vdash[10][n]+Vdash[11][n];
       OUTPUT <<n<<" "<<Vdash[n][1]<<" "<<Vdash[n][2]<<" "<<Vdash[n][3]<<" "<<Vdash[n][4]<<" "
                    <<Vdash[n][5]<<" "<<Vdash[n][6]<<" "<<Vdash[n][7]<<" "<<Vdash[n][8]<<" "
                    <<Vdash[n][9]<<" "<<Vdash[n][10]<<" "<<Vdash[n][11]<<"  "<<Osum[n]<<"\n";
       n=n+1;
      }while(n<=nZone);
       OUTPUT <<"Di "<<Dsum[1]<<" "<<Dsum[2]<<" "<<Dsum[3]<<" "<<Dsum[4]
               <<" "<<Dsum[5]<<" "<<Dsum[6]<<" "<<Dsum[7]<<" "<<Dsum[8]
               <<" "<<Dsum[9]<<" "<<Dsum[10]<<" "<<Dsum[11]<<"\n";
      itration=itration+1;
     }while(itration<=30);

return 0;
}
```

Table D.4 Input Data for the Trip Generation/Distribution Model (Persons).

Zone	1	2	3	4	5	6	7	8	9	10	11	Sum
O_i	2	7	16	23	59	25	11	19	2	27	1	192
D_i	8	9	18	14	59	18	19	17	9	15	7	193

Table D.5 Input Data for the Trip Generation/Distribution Model (Travel Time; Min.).

Zone	1	2	3	4	5	6	7	8	9	10	11
1	0.5	12	5.9	6.5	9.1	10.1	14	7.9	17.9	7.4	13.2
2	12	2.5	6.3	6.6	3.8	2.4	2.2	5.8	7.8	6.7	6.8
3	5.9	6.3	2.5	3.2	4.4	4.2	8.2	3.2	13.4	8.4	6.6
4	6.5	6.6	3.2	6.5	3.9	4.3	8.6	4.3	13.1	7.5	3.7
5	11.8	4.9	5.7	5.1	4	4.2	7.8	4.2	12.1	5.6	10.2
6	10.1	2.4	4.2	4.3	3.2	1.5	4.1	4.8	10.4	7.6	5
7	14	2.2	8.2	8.6	6	4.1	1.5	7.9	8	6.4	6.2
8	7.9	5.8	3.2	4.3	4.2	4.8	7.9	13	11.1	5.2	5.4
9	17.9	7.8	13.4	13.1	9.3	10.4	8	11.1	1.5	5.1	14.2
10	7.4	6.7	8.4	7.5	5.6	7.6	6.4	5.2	5.1	0.5	7.3
11	13.2	6.8	6.6	3.7	10.2	5	6.2	5.4	11.2	7.3	0.9

Table D.6 Trip Generation/Distribution Model Program Output (Persons).

Zone	1	2	3	4	5	6	7	8	9	10	11	O_i
1	1.14	0.03	0.13	0.10	0.25	0.06	0.07	0.10	0.04	0.05	0.03	2
2	0.16	0.51	0.41	0.34	2.05	0.88	1.50	0.45	0.33	0.18	0.20	7
3	0.84	0.51	2.61	1.81	4.50	1.28	1.02	2.06	0.49	0.37	0.53	16
4	1.18	0.75	3.15	1.38	7.86	1.93	1.51	2.37	0.77	0.63	1.47	23
5	1.81	2.84	4.94	4.90	21.40	5.52	4.64	6.77	2.34	2.37	1.49	59
6	0.62	1.71	1.97	1.71	7.88	4.55	2.60	1.75	0.80	0.51	0.89	25
7	0.24	0.99	0.54	0.46	2.25	0.89	3.80	0.57	0.56	0.33	0.39	11
8	0.86	0.76	2.80	1.85	6.50	1.54	1.46	0.70	0.81	0.81	0.90	19
9	0.05	0.07	0.09	0.08	0.38	0.09	0.19	0.11	0.78	0.11	0.04	2
10	1.03	0.74	1.20	1.19	5.48	1.09	2.03	1.96	1.99	9.53	0.74	27
11	0.03	0.03	0.07	0.11	0.14	0.08	0.10	0.09	0.04	0.03	0.28	1
D_i	8	9	18	14	59	18	19	17	9	15	7	192

D.3 MODSIM Program Code for BT Operation Simulation Model.

```
DEFINITION MODULE busim;
VAR
  Ch                   : CHAR;
  Alight               : REAL;
  Alighttime           : REAL;
  Accel                : REAL;
  Acceleration         : REAL;
  Altrt                : ARRAY INTEGER OF INTEGER;
  Arrival              : REAL;
  ATLOS                : REAL;
  AutoTT               : REAL;
  ATP                  : REAL;
  ATRELT               : REAL;
  ATU                  : REAL;
  Avggrade             : ARRAY INTEGER OF REAL;
  Baseyrinveh          : REAL;
  BasicResistance      : REAL;
  Bestmidpath          : INTEGER;
  Bestmidtotdist       : REAL;
  Boarding             : REAL;
  Boardingtime         : REAL;
  Btdemand             : REAL;
  BTLOS                : REAL;
  BTP                  : REAL;
  BTRELT               : REAL;
  BTU                  : REAL;
  Busnewton            : REAL;
  BusDeceleration      : REAL;
  BusPosition          : REAL;
  BusPositionTotal     : REAL;
  Buston               : REAL;
  BusVelocity          : REAL;
  Cap                  : ARRAY INTEGER, INTEGER OF REAL;
  Capacity             : REAL;
  Class                : ARRAY INTEGER OF INTEGER;
  CongestionRate       : REAL;
  Cycle                : REAL;
  CriticalGap          : REAL;
  d                    : ARRAY INTEGER OF INTEGER;
  d1                   : REAL;
  d2                   : REAL;
  Delay                : ARRAY INTEGER OF REAL;
  dist                 : ARRAY INTEGER, INTEGER OF REAL;
  distance             : REAL;
  dwelltime            : REAL;
  diameter             : REAL;
  dummy                : INTEGER;
  efficiency           : REAL;
  EnergyConsumed       : REAL;
  Energy               : REAL;
  Firstcome            : INTEGER;
  fleetsize            : INTEGER;
  gradesum             : REAL;
  grade                : REAL;
  green                : REAL;
  GravityTerm          : ARRAY INTEGER OF REAL;
  growthrate           : REAL;
  hp                   : ARRAY INTEGER OF REAL;
  horizonyr            : INTEGER;
```

```
i                   : INTEGER;
IniATP              : REAL;
inidemand           : REAL;
iniBTdemand         : REAL;
inifleetsize        : INTEGER;
initime             : INTEGER;
IntLine             : INTEGER;
Intsectype          : INTEGER;
invehtime           : ARRAY INTEGER, INTEGER OF INTEGER;
J                   : REAL;
j                   : INTEGER;
k                   : INTEGER;
l                   : INTEGER;
LOS                 : REAL;
m                   : INTEGER;
midpath             : ARRAY INTEGER OF INTEGER;
midtotdist          : ARRAY INTEGER OF REAL;
MinStopTime         : REAL;
mktpnt              : REAL;
nextstpndly         : REAL;
n                   : REAL;
nn                  : INTEGER;
NumOfRt             : INTEGER;
NumOfAltn           : INTEGER;
NumOperation        : REAL;
numyear             : INTEGER;
numsec              : ARRAY INTEGER OF INTEGER;
o                   : ARRAY INTEGER OF INTEGER;
origin              : REAL;
OTLOS               : REAL;
OTP                 : REAL;
OTRELT              : REAL;
OTU                 : REAL;
pcu                 : REAL;
PF                  : REAL;
PushPedal           : REAL;
pax                 : ARRAY INTEGER OF REAL;
paxMassNewton       : ARRAY INTEGER OF REAL;
previousstpndly     : REAL;
previouspax         : REAL;
previousboarding    : REAL;
previousalight      : REAL;
Qtotal              : ARRAY INTEGER OF REAL;
RealLine            : REAL;
route               : INTEGER;
RPM                 : ARRAY INTEGER OF REAL;
Rtcap               : ARRAY INTEGER OF REAL;
Sigmainvtt          : ARRAY INTEGER OF REAL;
spotnum             : INTEGER;
schdltime           : INTEGER;
stopdelay           : INTEGER;
stopnumb            : INTEGER;
stopdistnt          : ARRAY INTEGER OF REAL;
spotheight          : ARRAY INTEGER OF REAL;
stopheight          : ARRAY INTEGER OF REAL;
stopid              : ARRAY INTEGER OF INTEGER;
StopTime            : ARRAY INTEGER OF REAL;
spot                : ARRAY INTEGER, INTEGER OF REAL;
spots               : REAL;
startdist           : REAL;
StopDistance        : REAL;
```

```
  stpndly             : REAL;
  TE                  : ARRAY INTEGER, INTEGER OF REAL;
  time                : INTEGER;
  Tn                  : ARRAY INTEGER, INTEGER OF REAL;
  Tnini               : REAL;
  To                  : ARRAY INTEGER, INTEGER OF REAL;
  totalstopdist       : ARRAY INTEGER OF REAL;
  totaldist           : ARRAY INTEGER OF REAL;
  TotalMassNewton     : ARRAY INTEGER OF REAL;
  Totalposition       : REAL;
  Totalenergyco       : ARRAY INTEGER OF REAL;
  totalpax            : REAL;
  TotalResistance     : REAL;
  totalTn             : REAL;
  totinvehtime        : ARRAY INTEGER OF REAL;
  TractiveEffort      : REAL;
  trafficdemand       : REAL;
  TGear               : ARRAY INTEGER OF REAL;
  u                   : ARRAY INTEGER OF REAL;
  V                   : ARRAY INTEGER, INTEGER OF REAL;
  Vol                 : ARRAY INTEGER, INTEGER OF REAL;
  volume              : REAL;
  X                   : ARRAY INTEGER OF REAL;
  year                : INTEGER;
PROCEDURE busim1;
END MODULE.
(*****************************************************************************)
(*****************************************************************************)
IMPLEMENTATION MODULE busim;
(** program 6 : get the natural trend of tt and fuel when considering natural increase
of pop. according to the B'burg office**)
FROM IOMod IMPORT ReadKey, StreamObj, ALL FileUseType;
FROM MathMod IMPORT pi, SQRT, ATAN, EXP, CEIL;
VAR
  Strm : StreamObj;
  outStrm : StreamObj;
PROCEDURE busim1;
BEGIN
NEW (Strm);
(*******************************  SPECIFICATINS  *****************************)
ASK Strm TO Open ("d:\0integ\integrat\data\buschar.dat", Input);
(*****************************************************************************)
horizonyr:=2020;
(*****************************************************************************)
(*******************************  SPECIFICATINS  *****************************)
NEW ( u, 1..5);
NEW ( RPM, 1..5);
NEW ( hp, 1..5);
NEW ( V, 1..5, 1..5);
NEW ( TE, 1..5, 1..5);
       FOR i :=1 TO 5
       ASK Strm TO ReadReal(RealLine);
       u[i]:=RealLine;
       END FOR;
       ASK Strm TO ReadReal(RealLine);
       J:=RealLine;
       FOR i :=1 TO 5
       ASK Strm TO ReadReal(RealLine);
       RPM[i]:=RealLine;
       END FOR;
       FOR i :=1 TO 5
       ASK Strm TO ReadReal(RealLine);
       hp[i]:=RealLine;
       END FOR;
       ASK Strm TO ReadReal(RealLine);
       diameter:=RealLine;
       ASK Strm TO ReadReal(RealLine);
       efficiency:=RealLine;
       ASK Strm TO ReadReal(RealLine);
       buston:=RealLine;
       busnewton:=buston*1000./9.81;
       n:=1.;
       FOR i :=1 TO 5
              FOR j :=1 TO 5
              V[i, j]:=(RPM[i]*diameter*pi/( J*u[j]))*(60./1000.);
              TE[i, j]:=2650.*hp[i]*efficiency/V[i, j];
              n:=n+1.;
              END FOR;
       END FOR;
ASK Strm TO Close;
DISPOSE (Strm);
(******************************FROM IPAX.MOD**********************************)
Totalposition:=0.;
numyear:=horizonyr-1997;
NEW (invehtime, 1..numyear, 1..7);
NEW (totinvehtime, 1..numyear);
NEW (Totalenergyco, 1..numyear);
growthrate:=1.;
mktpnt:=0.;
NumOperation:=5.;
IniATP:=.6965;(** Initial ATP in 1998 =.6965**)
ATP:=IniATP;
NEW (outStrm);
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1allrt.out", Output);
(*****************************************************************************)
(*****************************************************************************)
FOR year:= 1 TO numyear;
(*****************************************************************************)
(*****************************************************************************)
(**OUTPUT("START work yr=                  ",year+1997);**)
NEW (Strm);
(******************************  SPECIFICATINS  ******************************)
NumOfRt:=7;
NumOfAltn:=3;
(******************************  SPECIFICATINS  ******************************)
NEW (o, 1..NumOfRt);
NEW (altrt, 1..NumOfAltn);
NEW (To, 1..NumOfRt, 1..NumOfAltn);
NEW (Vol, 1..NumOfRt, 1..NumOfAltn);
NEW (Cap, 1..NumOfRt, 1..NumOfAltn);
NEW (Tn, 1..NumOfRt, 1..NumOfAltn);
NEW (dist, 1..NumOfRt, 1..NumOfAltn);
NEW (Sigmainvtt, 1..NumOfRt);
NEW (Qtotal, 1..NumOfRt);
NEW (Rtcap, 1..NumOfRt);
ASK Strm    TO Open ("d:\0integ\integrat\data\bpr.dat", Input);
(** DISCRIPTION OF DATA : 'ORIGIN NODE','ALTERNATIVE ROUTE','DISTANCE',
                          'FREE FLOW TRAVEL TIME','VOLUME','CAPACITY'.**)
IF (1<year) AND (year<4)                (** 1999 - 2000 **)
       growthrate:=growthrate*1.0098;          (** YEARLY GROWTH RATE OF PAX.**)
       trafficdemand:=inidemand*growthrate;
ELSIF (4<=year) AND (year<14)           (** 2001 - 2010 **)
```

```
       growthrate:=growthrate*1.0077;          (** YEARLY GROWTH RATE OF PAX.**)
       trafficdemand:=inidemand*growthrate;
ELSIF (14<=year)                  (** 2011 -      **)
       growthrate:=growthrate*1.0058;          (** YEARLY GROWTH RATE OF PAX.**)
       trafficdemand:=inidemand*growthrate;
END IF;
FOR i :=1 TO NumOfRt
FOR j :=1 TO NumOfAltn
       ASK Strm TO ReadInt(IntLine);
        o[i]:=i;
       ASK Strm TO ReadInt(IntLine);
        altrt[j]:=j;
       ASK Strm TO ReadReal(RealLine);
        dist[i,j]:=RealLine;
       ASK Strm TO ReadReal(RealLine);
        To[i,j]:=RealLine;
       ASK Strm TO ReadReal(RealLine);
        Vol[i,j]:=RealLine;
IF year<=2
Vol[i,j]:=(Vol[i,j]+Vol[i,j]*(growthrate-1.)/1.2);
ELSE
Vol[i,j]:=(Vol[i,j]+Vol[i,j]*(growthrate-1.)/1.2)*ATP/IniATP;
END IF;
       Qtotal[i]:=Qtotal[i]+Vol[i,j];
       ASK Strm TO ReadReal(RealLine);
        Cap[i,j]:=RealLine;
END FOR;
         Rtcap[i]:=Cap[i,1];
END FOR;                   (** End Read Data **)
ASK Strm TO Close;
DISPOSE (Strm);
FOR k :=1 TO 7                    (** Start BPR & Traffic Assignment **)
totalTn:=0.;
FOR i :=1 TO NumOfRt
FOR j :=1 TO NumOfAltn
       LOS:=Vol[i,j]/Cap[i,j];
              IF (k=7) AND (i=1) AND (j=1)
              OUTPUT("LOS =", LOS);
              END IF;
       Tn[i,j]:=To[i,j]*(1.+0.15*(LOS*LOS*LOS*LOS));
END FOR;
        totalTn:=totalTn+Tn[i,1];
END FOR;
IF year=1;
   Tnini:=totalTn;
END IF;
FOR i :=1 TO NumOfRt
FOR j :=1 TO NumOfAltn
       Sigmainvtt[i]:=Sigmainvtt[i]+(1./Tn[i,j]);
       (**OUTPUT("Sigmainvtt(",i,") = ",Sigmainvtt[i]);**)
END FOR;
END FOR;
FOR i :=1 TO NumOfRt
FOR j :=1 TO NumOfAltn
       Vol[i,j]:=Qtotal[i]*(1./Tn[i,j])/Sigmainvtt[i];
       (**OUTPUT("Vol(",i,",",j,") = ",Vol[i,j]);**)
END FOR;
       Sigmainvtt[i]:=0.;
END FOR;
END FOR;            (** End BPR & Traffic Assignment **)
        CongestionRate:=totalTn/Tnini;
       AutoTT:=9.11*CongestionRate;
OUTPUT("AutoTT = ",AutoTT);
OUTPUT("CongestionRate,totalTn,Tnini= ",CongestionRate," ",totalTn," ",Tnini);
(**
NEW (outStrm);
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1bpr.out", Output);
ASK outStrm TO WriteString("In-Veh. Auto Travel time (1998=9.11min) = ");
ASK outStrm TO WriteReal(AutoTT,10,3);
ASK outStrm TO WriteLn;
ASK outStrm TO Close;
DISPOSE (outStrm)**)

(******************** MARKET PENETRATION & MODE CHOICE ************************)
IF (2=year)
       mktpnt:= .3;                     (** YEARLY GROWTH RATE OF Mktpnt**)
ELSIF (2<year) AND (year<7)
       mktpnt:= mktpnt*1.1362;          (** YEARLY GROWTH RATE OF Mktpnt**)
ELSIF (7<=year) AND (year<12)
       mktpnt:= mktpnt*1.0845;          (** YEARLY GROWTH RATE OF Mktpnt**)
ELSIF (12<=year) AND (year<17)
       mktpnt:= mktpnt*1.0371;          (** YEARLY GROWTH RATE OF Mktpnt**)
ELSIF (17<=year)
       mktpnt:= mktpnt*1.0078;          (** YEARLY GROWTH RATE OF Mktpnt**)
END IF;
(******************************************************************************)
(******************************************************************************)
 mktpnt:= 0.;       (**< -------------------- option 1!! **)
(******************************************************************************)
(******************************************************************************)
OUTPUT(" mktpnt=",mktpnt);
IF 1<year
ATRELT:= totalTn/(totalTn*.5+231.*.55);
(**         ATRELT = Inveh/(Inveh+(access+egress) ***)
BTRELT:= totinvehtime[year-1]/((5.79-2.52*mktpnt)*432.+totinvehtime[year-1]+4350.);
 (**  BTRELT = In-veh/(wait(considered market share)+inveh+(access+egress) ***)
OTRELT:=.9;
ATLOS:=.8*2.7+.2*2.7*(Tnini/totalTn);
 (**   ATLOS is 20% proportional to the street travel time;                ***)
(**   other 80% is proportional to the LOS of parking, facilities, etc.    ***)
BTLOS:=5.19+.54*(mktpnt+(1.-mktpnt)*FLOAT(fleetsize-inifleetsize)/10.);
(**  BTLOS = when market penetration is 100%, the LOS will be 5.73<--survey **)
(**BTLOSisconsitedofmktpnt&(fleetsize of (1-mktpnt));                       **)
(**         assumption : fleetsize17 = no wait time                         **)
(**OUTPUT(" BTLOS=",BTLOS);                                                 **)
OTLOS:=2.85;
ATU:= 1.758*ATRELT+.667*ATLOS+2.205;
BTU:= 1.758*BTRELT+.667*BTLOS-1.129;
OTU:= 1.758*OTRELT+.667*OTLOS;
IF year=1
IniATP:=ATP;
END IF;
ATP:= EXP(ATU)/(EXP(ATU)+EXP(BTU)+EXP(OTU));
BTP:= EXP(BTU)/(EXP(ATU)+EXP(BTU)+EXP(OTU));
OTP:= EXP(OTU)/(EXP(ATU)+EXP(BTU)+EXP(OTU));
(**
    ASK outStrm TO WriteInt(year+1997,4);
    ASK outStrm TO WriteReal(ATLOS,7,4);
    ASK outStrm TO WriteReal(BTLOS,7,4);
    ASK outStrm TO WriteReal(OTLOS,7,4);
    ASK outStrm TO WriteReal(ATRELT,7,4);
    ASK outStrm TO WriteReal(BTRELT,7,4);
    ASK outStrm TO WriteReal(OTRELT,7,4);
    ASK outStrm TO WriteReal(ATP,7,4);
    ASK outStrm TO WriteReal(BTP,7,4);
    ASK outStrm TO WriteReal(OTP,7,4);
    ASK outStrm TO WriteLn;
**)
END IF;
(*****************************************************************************)
(*****************************************************************************)
time:=0;
totinvehtime[year]:=0.;
Totalenergyco[year]:=0.;
previouspax:=0.;
previousalight:=0.;
totalpax:=0.;
BusPositionTotal:=0.;
(*****************************************************************************)
(*****************************************************************************)
FOR route:= 1 TO 7;
NEW (Strm);
alighttime:=0.;
Accel:=0.;
Acceleration:=0.;
arrival:=0.;
BasicResistance:=0.;
BusDeceleration:=0.;
BusVelocity:=0.;
CriticalGap:=0.;
Energy:=0.;
gradesum:=0.;
grade:=0.;
initime:=0;
d1:=0.;
d2:=0.;
nextstpndly:=0.;
invehtime[year,route]:=0;
firstcome:=0;
(******************************  SPECIFICATINS  ******************************)
IF route = 1
       ASK Strm TO Open ("d:\0integ\integrat\data\pax31.dat", Input);
        spotnum  :=88;
       stopnumb :=16;
        stopdelay:=10;
        pcu      :=1.6;
        PushPedal:=1.-(.3*CongestionRate);
        schdltime:=900;
ELSIF route = 2
       ASK Strm TO Open ("d:\0integ\integrat\data\pax32.dat", Input);
        spotnum  :=58;
       stopnumb :=14;
        stopdelay:=9;
        schdltime:=1800;
ELSIF route = 3
       ASK Strm TO Open ("d:\0integ\integrat\data\pax33.dat", Input);
        spotnum  :=44;
       stopnumb := 6;
        stopdelay:=5;
        PushPedal:=1.-(.65*CongestionRate);
        schdltime:=2400;
ELSIF route = 4
       ASK Strm TO Open ("d:\0integ\integrat\data\pax34.dat", Input);
```

```
        spotnum  :=98;
       stopnumb :=17;
        stopdelay:=9;
        PushPedal:=1.-(.3*CongestionRate);
        schdltime:=3300;
ELSIF route = 5
       ASK Strm TO Open ("d:\0integ\integrat\data\pax35.dat", Input);
        spotnum  :=70;
       stopnumb := 8;
        stopdelay:=8;
        schdltime:=3900;
ELSIF route = 6
       ASK Strm TO Open ("d:\0integ\integrat\data\pax36.dat", Input);
        spotnum  :=53;
       stopnumb := 8;
        stopdelay:= 5;
        schdltime:=4500;
ELSE
       ASK Strm TO Open ("d:\0integ\integrat\data\pax37.dat", Input);
        spotnum  :=99;
       stopnumb :=23;
        stopdelay:=9;
        schdltime:=5400;
END IF;
NEW ( pax, 1..stopnumb);
NEW ( stopid, 0..stopnumb);
NEW ( paxMassNewton, 1..stopnumb);
NEW ( StopTime, 1..stopnumb);
NEW ( TotalMassNewton, 1..stopnumb);
  MinStopTime:=0.;                (**  ASSUMPTION  **)
       BTdemand:=trafficdemand*BTP;
       ASK Strm TO ReadInt(IntLine);
       dummy:=IntLine;
       ASK Strm TO ReadReal(RealLine);
       previousboarding:=RealLine;
IF year>1
       previousboarding:=BTdemand*previousboarding/iniBTdemand;
END IF;
   FOR i :=1 TO stopnumb (** # of stops**)
       ASK Strm TO ReadInt(IntLine);
       stopid[i]:=IntLine;
       ASK Strm TO ReadReal(RealLine);
       boarding:=RealLine;
       ASK Strm TO ReadReal(RealLine);
       alight:=RealLine;
       IF year>1
       boarding:=BTdemand*boarding/iniBTdemand;
       alight:=BTdemand*alight/iniBTdemand;
       END IF;
       totalpax:=totalpax+alight;
       pax[i]:=previouspax+previousboarding-previousalight;
       paxMassNewton[i]:=pax[i]*800./9.81;
       TotalMassNewton[i]:=paxMassNewton[i]+busnewton;
              IF (boarding=0.) AND (alight=0.)
       boardingtime:=0.;
       alighttime:=  0.;
              ELSE
       boardingtime:=4.1+boarding*3.;
       alighttime:=  4.1+alight  *3.;
              END IF;
       StopTime[i]:=MAXOF(MinStopTime, boardingtime, alighttime);
       previouspax:=pax[i];
       previousboarding:=boarding;
       previousalight:=alight;
   END FOR;
ASK Strm TO Close;
DISPOSE (Strm);
(******************************FROM IGGPH.MOD*********************************)
NEW (Strm);
(******************************  SPECIFICATINS  ******************************)
IF    route = 1
       ASK Strm TO Open ("d:\0integ\integrat\data\spot31.dat", Input);
ELSIF route = 2
       ASK Strm TO Open ("d:\0integ\integrat\data\spot32.dat", Input);
ELSIF route = 3
       ASK Strm TO Open ("d:\0integ\integrat\data\spot33.dat", Input);
ELSIF route = 4
       ASK Strm TO Open ("d:\0integ\integrat\data\spot34.dat", Input);
ELSIF route = 5
       ASK Strm TO Open ("d:\0integ\integrat\data\spot35.dat", Input);
ELSIF route = 6
       ASK Strm TO Open ("d:\0integ\integrat\data\spot36.dat", Input);
ELSIF route = 7
       ASK Strm TO Open ("d:\0integ\integrat\data\spot37.dat", Input);
END IF;
(******************************  SPECIFICATINS  ******************************)
NEW (totalstopdist, -1..stopnumb);
NEW (stopdistnt, 1..spotnum);
NEW (avggrade, 0..100);
NEW (totaldist, 0..spotnum);
NEW (class, 0..spotnum);
NEW (spot, 0..spotnum, 1..2);
NEW (spotheight, -1..spotnum);
NEW (stopheight, 0..100);
NEW (GravityTerm, 0..100);
m:=0;
origin:=0.;
FOR i:=0 TO spotnum
ASK Strm TO ReadReal(RealLine);
spot[i,1]:=RealLine;
ASK Strm TO ReadReal(RealLine);
spot[i,2]:=RealLine;
ASK Strm TO ReadInt(IntLine);
dummy:=IntLine;
ASK Strm TO ReadInt(IntLine);
class[i]:=IntLine;
ASK Strm TO ReadReal(RealLine);
totaldist[i]:=RealLine;
ASK Strm TO ReadReal(RealLine);
spotheight[i]:=RealLine;
  IF (class[i] <> 999) AND (class[i]<>0)
m:=m+1;
arrival:=totaldist[i];
stopdistnt[i]:=totaldist[i]-origin;
origin:=arrival;
  IF i=0
grade:=0.;
avggrade[m]:=0.;
  ELSE
grade:=(spotheight[i]-spotheight[i-1])*100./(totaldist[i]-totaldist[i-1]);
avggrade[m]:=(gradesum + grade) / ( spots + 1.);
  END IF;
```

```
spots:=0.;
gradesum:=0.;
  ELSIF (class[i]<>0)
stopdistnt[i]:=0.;
grade:=(spotheight[i]-spotheight[i-1])*100./(totaldist[i]-totaldist[i-1]);
gradesum:=gradesum+grade;
spots:=spots+1.;
  ELSE
dummy:=dummy;
  END IF;
END FOR;
(******************************FROM IGGPH2.MOD********************************)
       j:=1;
       k:=1;
FOR i:=1 TO spotnum
GravityTerm[i]:=TotalMassNewton[j]*ATAN(avggrade[k]/100.);
(**OUTPUT(time," avggrade[k]=",avggrade[k]);**)
IF class[i]=999
       dummy:=dummy;
ELSIF class[i]>999
       k:=k+1;
ELSE
       k:=k+1;
       j:=class[i]+1;
END IF;
END FOR;
ASK Strm TO Close;
DISPOSE (Strm);
(******************************FROM ISTOPDLY.MOD******************************)
NEW (Strm);
(******************************  SPECIFICATINS  ******************************)
IF    route = 1
       ASK Strm TO Open ("d:\0integ\integrat\data\intsec31.dat", Input);
ELSIF route = 2
       ASK Strm TO Open ("d:\0integ\integrat\data\intsec32.dat", Input);
ELSIF route = 3
       ASK Strm TO Open ("d:\0integ\integrat\data\intsec33.dat", Input);
ELSIF route = 4
       ASK Strm TO Open ("d:\0integ\integrat\data\intsec34.dat", Input);
ELSIF route = 5
       ASK Strm TO Open ("d:\0integ\integrat\data\intsec35.dat", Input);
ELSIF route = 6
       ASK Strm TO Open ("d:\0integ\integrat\data\intsec36.dat", Input);
ELSIF route = 7
       ASK Strm TO Open ("d:\0integ\integrat\data\intsec37.dat", Input);
END IF;
(******************************  SPECIFICATINS  ******************************)
NEW ( numsec, 1..spotnum);
NEW ( X, 1..stopdelay);
NEW ( Delay, 1..stopdelay);
       FOR i :=1 TO stopdelay
       ASK Strm TO ReadInt(IntLine);
       numsec[i]:=IntLine;
       ASK Strm TO ReadInt(IntLine);
       Intsectype:=IntLine;
       IF (Intsectype=1001) OR (Intsectype=1002) OR (Intsectype=1003)
          ASK Strm TO ReadReal(RealLine);
          PF:=RealLine;
          ASK Strm TO ReadReal(RealLine);
          cycle:=RealLine;
          ASK Strm TO ReadReal(RealLine);
          green:=RealLine;
          ASK Strm TO ReadReal(RealLine);
          volume:=RealLine;
          IF year>=2
          volume:=(volume+volume*(growthrate-1.)/1.2)*ATP/IniATP;
          END IF;
          ASK Strm TO ReadReal(RealLine);
          capacity:=RealLine;
          X[i]:=volume/capacity;
d1:=.38*cycle*((1.-green/cycle)*(1.-green/cycle)/(1.-((green/cycle)*X[i])));
d2:=173.*X[i]*X[i]*((X[i]-1.)+SQRT((X[i]-1.)*(X[i]-1.)+(16.*X[i]/capacity)));
          Delay[i]:=PF*(d1+d2);
       ELSIF Intsectype=2004
          ASK Strm TO ReadReal(RealLine);
          volume:=RealLine;
          IF year>=2
          volume:=(volume+volume*(growthrate-1.)/1.2)*ATP/IniATP;
          END IF;
          ASK Strm TO ReadReal(RealLine);
          capacity:=RealLine;
          Delay[i]:=EXP(3.8*(volume/capacity));
       ELSIF  Intsectype=20021 (**RT from Major**)
             Delay[i]:=3.;
       ELSE
           IF Intsectype=20022 (**RT from Minor to Major**)
             ASK Strm TO ReadReal(RealLine);
             volume:=RealLine;
             IF year>=2
             volume:=(volume+volume*(growthrate-1.)/1.2)*ATP/IniATP;
             END IF;
             CriticalGap:=6.*pcu*CongestionRate;
           ELSIF Intsectype=20031 (**LT from 2 lane Major**)
             ASK Strm TO ReadReal(RealLine);
             volume:=RealLine;
             IF year>=2
             volume:=(volume+volume*(growthrate-1.)/1.2)*ATP/IniATP;
             END IF;
             CriticalGap:=5.25*pcu*CongestionRate;
           ELSIF Intsectype=20032 (**LT from 4 lane Major**)
             ASK Strm TO ReadReal(RealLine);
             volume:=RealLine;
             IF year>=2
             volume:=(volume+volume*(growthrate-1.)/1.2)*ATP/IniATP;
             END IF;
             CriticalGap:=5.75*pcu*CongestionRate;
           ELSIF Intsectype=20033 (**LT from Minor to 2 lane Major**)
             ASK Strm TO ReadReal(RealLine);
             volume:=RealLine;
             IF year>=2
             volume:=(volume+volume*(growthrate-1.)/1.2)*ATP/IniATP;
             END IF;
             CriticalGap:=7.25*pcu*CongestionRate;
           ELSIF Intsectype=20034 (**LT from Minor to 4 lane Major**)
             ASK Strm TO ReadReal(RealLine);
             volume:=RealLine;
             IF year>=2
             volume:=(volume+volume*(growthrate-1.)/1.2)*ATP/IniATP;
             END IF;
             CriticalGap:=7.75*pcu*CongestionRate;
           ELSE
             OUTPUT("ERROR IN IMPLEMENTATION MODULE stopdly !!!!");
           END IF;
          Delay[i]:=(3600./volume)*(EXP((volume/3600.)*CriticalGap)-1.-
(volume/3600.)*CriticalGap);
       END IF;
       END FOR;
ASK Strm TO Close;
DISPOSE (Strm);
(*****************************FROM ISIMTN.MOD*********************************)
NEW (TGear, 1..5);
(***************************  SPECIFICATINS  *********************************)
(**    IF    route = 1
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1r1.out", Output);
       ELSIF route = 2
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1r2.out", Output);
       ELSIF route = 3
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1r3.out", Output);
       ELSIF route = 4
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1r4.out", Output);
       ELSIF route = 5
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1r5.out", Output);
       ELSIF route = 6
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1r6.out", Output);
       ELSE
       ASK outStrm TO Open ("d:\0integ\integrat\output\pn1r7.out", Output);
       END IF;
**)
(***************************  SPECIFICATINS  *********************************)
stpndly:=0.;
previousstpndly:=0.;
BusPosition:=0.;
EnergyConsumed:=0.;
(**
ASK outStrm TO WriteString("# ROUTE = ");
ASK outStrm TO WriteInt(route,2);
ASK outStrm TO WriteLn;
ASK outStrm TO WriteString(" T I M E //   BUS   //    BUS   //    FUEL    //");
ASK outStrm TO WriteLn;
ASK outStrm TO WriteString("(sec) // VELOCITY // POSITION // CONSUMPTION  //");
ASK outStrm TO WriteLn;
ASK outStrm TO WriteString("      // (m/sec2) //    (m)   //      (g)     //");
ASK outStrm TO WriteLn;
**)
distance:=0.;
m:=1;
nn:=1;
i:=1;
k:=1;
FOR k:=1 TO spotnum
dwelltime:=0.;
startdist:=0.;
IF    class[k]<>999
          distance:= totaldist[k];
       IF    class[k] = 1001                   (** for signal straight  **)
                IF k <> numsec[m]
                    stpndly:=1.;        (**asumption**)
                ELSE
                    stpndly:=Delay[m];
                    m:=m+1;
                END IF;
(**
ASK outStrm TO WriteString("   Spot[");
ASK outStrm TO WriteInt(k,2);
ASK outStrm TO WriteString("] is signalized Stright turn; DELAY =");
ASK outStrm TO WriteReal(stpndly,9,3);
ASK outStrm TO WriteLn;
**)
       ELSIF class[k] = 1002                   (** for signal right turn **)
                IF k <> numsec[m]
                    stpndly:=2.;        (**asumption**)
                ELSE
                    stpndly:=Delay[m];
                    m:=m+1;
                END IF;
(**
ASK outStrm TO WriteString("   Spot[");
ASK outStrm TO WriteInt(k,2);
ASK outStrm TO WriteString("] is signalized right turn; DELAY =");
ASK outStrm TO WriteReal(stpndly,9,3);
ASK outStrm TO WriteLn;
**)
       ELSIF class[k] = 1003                   (** for signal left turn **)
                IF k <> numsec[m]
                    stpndly:=5.;        (**asumption**)
                ELSE
                    stpndly:=Delay[m];
                    m:=m+1;
                END IF;
(**
ASK outStrm TO WriteString("   Spot[");
ASK outStrm TO WriteInt(k,2);
ASK outStrm TO WriteString("] is signalized left turn; DELAY =");
ASK outStrm TO WriteReal(stpndly,9,3);
ASK outStrm TO WriteLn;
**)
       ELSIF class[k] = 2002                   (** for no- signal right turn **)
                IF k <> numsec[m]
                    stpndly:=4.;        (**asumption**)
                ELSE
                    stpndly:=Delay[m];
                    m:=m+1;
                END IF;
(**
ASK outStrm TO WriteString("   Spot[");
ASK outStrm TO WriteInt(k,2);
ASK outStrm TO WriteString("] is unsignalied right turn; DELAY =");
ASK outStrm TO WriteReal(stpndly,9,3);
ASK outStrm TO WriteLn;
**)
       ELSIF class[k] = 2003                   (** for no- signal left turn **)
                IF k <> numsec[m]
                    stpndly:=5.;        (**asumption**)
                ELSE
                    stpndly:=Delay[m];
                    m:=m+1;
                END IF;
(**
ASK outStrm TO WriteString("   Spot[");
ASK outStrm TO WriteInt(k,2);
ASK outStrm TO WriteString("] is unsignalied left turn; DELAY =");
ASK outStrm TO WriteReal(stpndly,9,3);
ASK outStrm TO WriteLn;
**)
```

```
       ELSE                                    (** for stops **)
                    i:=class[k];
                IF  stopid[i]=k
                    stpndly:=StopTime[nn];
                    nn:=nn+1;
                ELSE
                    ASK outStrm TO WriteString("ERROR!!!!!");
                END IF;
(**
ASK outStrm TO WriteString("   Spot[");
ASK outStrm TO WriteInt(k,2);
ASK outStrm TO WriteString("] is Stop;             STOP DEALY =");
ASK outStrm TO WriteReal(stpndly,9,3);
ASK outStrm TO WriteLn;
**)
       END IF;
       nextstpndly:=stpndly;
       stpndly:=previousstpndly;
             WHILE BusPosition<=distance
WHILE firstcome=0
initime:=time;
firstcome:=1;
END WHILE;
       time:=time+1;
       IF (dwelltime<=stpndly)
             BusVelocity:=0.;
             dwelltime:=dwelltime+1.;
IF time<200
END IF;
       ELSE
             IF BusPosition>(distance-StopDistance)
                    Accel:=-1.*BusDeceleration;
             ELSE
                    Accel:=Acceleration;
             END IF;
       BasicResistance:=960.+8.4*BusVelocity+.1403*BusVelocity*BusVelocity;
             IF BusVelocity<.0
                    BusDeceleration:=.0;
             ELSE
                    BusDeceleration:=1.;
             END IF;
                    TotalResistance:=BasicResistance+GravityTerm[k];
             IF (TractiveEffort-TotalResistance)>0.
             Acceleration:=(TractiveEffort-TotalResistance)/(TotalMassNewton[i]);
             ELSE
                    Acceleration:=0.;
             END IF;
             IF BusVelocity>.0
                    StopDistance:=BusVelocity*BusVelocity/(2.*BusDeceleration);
             ELSE
                    StopDistance:=.0;
             END IF;
       IF (dwelltime<=stpndly)
             BusVelocity:=0.;
       ELSE
             BusVelocity:=MAXOF(0.,BusVelocity+Accel);
       END IF;
             END IF;
             IF BusVelocity<15./3.6
                    TractiveEffort:=TGear[1]*PushPedal;
                    Energy:=160.*hp[1]*.59/3600.;
             ELSIF BusVelocity<30./3.6
                    TractiveEffort:=TGear[2]*PushPedal;
                    Energy:=145.*hp[2]*.59/3600.;
             ELSIF BusVelocity<45./3.6
                    TractiveEffort:=TGear[3]*PushPedal;
                    Energy:=130.*hp[3]*.59/3600.;
             ELSIF BusVelocity<60./3.6
                    TractiveEffort:=TGear[4]*PushPedal;
                    Energy:=150.*hp[4]*.59/3600.;
             ELSE
                    TractiveEffort:=TGear[5]*PushPedal;
                    Energy:=160.*hp[5]*.59/3600.;
             END IF;
             FOR j:=1 TO 5
             IF BusVelocity<V[j,2]
       TGear[j]:=(TE[j,2]-TE[j,1])*(BusVelocity-V[j,1])/(V[j,2]-V[j,1])+TE[j,1];
             ELSIF BusVelocity<V[j,3]
       TGear[j]:=(TE[j,3]-TE[j,2])*(BusVelocity-V[j,2])/(V[j,3]-V[j,2])+TE[j,2];
             ELSIF BusVelocity<V[j,4]
       TGear[j]:=(TE[j,4]-TE[j,3])*(BusVelocity-V[j,3])/(V[j,4]-V[j,3])+TE[j,3];
             ELSE
       TGear[j]:=(TE[j,5]-TE[j,4])*(BusVelocity-V[j,4])/(V[j,5]-V[j,4])+TE[j,4];
             END IF;
             END FOR;
             BusPosition:=BusPosition+BusVelocity;
             BusPositionTotal:=BusPositionTotal+BusVelocity;
             EnergyConsumed:=EnergyConsumed+Energy;
IF (year = 1) OR (year = 3) OR (year = 13) OR (year = 23)
       ASK outStrm TO WriteInt(time,4);
       ASK outStrm TO WriteReal(BusPositionTotal,15,3);
       ASK outStrm TO WriteLn;
END IF;
       END WHILE;
       BusVelocity:=0.;
IF k=spotnum (** aa**)
WHILE firstcome=1
invehtime[year,route]:=time-initime;
totinvehtime[year]:=totinvehtime[year]+FLOAT(invehtime[year,route]);
firstcome:=13;
END WHILE;
WHILE time<schdltime
Energy:=100.*hp[1]*.59/3600.;
EnergyConsumed:=EnergyConsumed+Energy;
time:=time+1;
IF (year = 1) OR (year = 3) OR (year = 13) OR (year = 23)
       ASK outStrm TO WriteInt(time,4);
       ASK outStrm TO WriteReal(BusPositionTotal,15,3);
       ASK outStrm TO WriteLn;
END IF;
END WHILE;
ELSE
       dummy:=dummy;
END IF; (** aa**)
ELSE
      dummy:=dummy;
END IF;
      dummy:=dummy+1;
       previousstpndly:=nextstpndly;
END FOR;
Totalposition:= Totalposition+BusPosition;
Totalenergyco[year]:= Totalenergyco[year] + EnergyConsumed;
```

```
(** OPTION I **)
(**
IF route=1
    ASK outStrm TO WriteString("YEAR //Route//TotalPax// In-veh.T.// Tot.In-veh.T.//
Fuel   //");
    ASK outStrm TO WriteLn;
END IF;
ASK outStrm TO WriteInt(year+1997,4);
ASK outStrm TO WriteInt(route,5);
ASK outStrm TO WriteReal(totalpax,11,1);
ASK outStrm TO WriteInt(invehtime[year,route],11);
ASK outStrm TO WriteReal(totinvehtime[year],14,1);
ASK outStrm TO WriteReal(Totalenergyco[year],15,2);
ASK outStrm TO WriteLn;
**)
END FOR;                 (**route** **route** **route** **route** **route**)
(**
ASK outStrm TO WriteLn;
ASK outStrm TO WriteInt(year+1997,4);
(**ASK outStrm TO WriteReal(BTdemand,11,4);**)
ASK outStrm TO WriteReal(LOS,11,4);
ASK outStrm TO WriteReal(mktpnt,11,4);
ASK outStrm TO WriteReal(totalTn,11,4);
ASK outStrm TO WriteReal(totalpax,11,4);
**)
(**OUTPUT("END   work yr=                 ",year+1997);**)
IF year=1
iniBTdemand:=totalpax;
inidemand:=totalpax/.1154; (**Initially BT had 11.54% of total demand**)
trafficdemand:=inidemand;
END IF;
fleetsize:=CEIL(1.5*(totalpax*NumOperation/60.)/3.);
(**fleetsize:=f(round trip;1.5hr.&bus:141&trooper:92,&pax;38seat;15standee; **)
(**  observed peak hr;3hrs.&5operations during pk.hrs.)                     **)
IF year=1
inifleetsize:=fleetsize;
END IF;
OUTPUT(year+1997,"   totalpax=",totalpax," fleetsize = ",fleetsize);
END FOR;
(** OPTION II **)
NEW (outStrm);
    ASK outStrm TO Open ("d:\0integ\integrat\output\pn1summ.out", Output);
FOR year:= 1 TO numyear;
    ASK outStrm TO WriteInt(year+1997,4);
    ASK outStrm TO WriteString("  Tot. In-Veh. TT = ");
    ASK outStrm TO WriteReal(totinvehtime[year],6,1);
    ASK outStrm TO WriteString("  Tot. Fuel       = ");
    ASK outStrm TO WriteReal(Totalenergyco[year],9,3);
    ASK outStrm TO WriteLn;
END FOR;
ASK outStrm TO Close;
DISPOSE (outStrm);
OUTPUT("HIT KEY");
ch:=ReadKey;
END PROCEDURE;
END MODULE.
```

Table D.7 Index of the Simulation Model Variables.

Variable	Description	Unit	Value[1]
alight	number of passengers alighting at each bus stop	persons	
alighttime	alight time on each bus stop	s	
Acceleration	Bus acceleration	*m*/s^2	
Altrt[*i*]	alternative route for auto on section *i*	dim	
arrival	arrival time to time-check	s	
ATLOS	level of service of Auto	dim	
AutoTT	travel time of auto	s	
ATRELT	relative time (in-vehicle time / total journey time) of auto	dim	
ATP	probability to select auto	dim	
ATU	Utility of Auto	dim	
avggrade[*i*]	average grade profile between two spots *i*–1 and *i*	dim	
baseyrinveh	base year (1998) in-vehicle travel time of Auto	s	
BasicResistance	Basic resistance of Bus	newton	
boarding	number of passengers boarding at each bus stop	persons	
boardingtime	boarding time at each bus stop	s	
BTdemand	passenger demand for BT (persons)	persons	
BTLOS	level of service of BT	dim	
BTP	probability to select BT	dim	
BTRELT	relative time (in-vehicle time / total journey time) of BT	dim	
BTU	Utility of BT	dim	
busnewton	BT empty weight on Newton	newton	
BusDeceleration	Deceleration late of BT speed at each time interval	*m*/s^2	
BusPosition	BT position along the study section	meters	
BusPositionTotal	BT position along the study route	meters	
buston	BT empty weight on Ton	ton	
BusVelocity	BT velocity at each time interval	*m*/s	
Cap[*i*,*j*]	Estimated capacity on section *i*, alternative route *j* for traffic assignment	*Veh/hr.*	
capacity	Capacity of each intersection along the BT route	*Veh/hr.*	
class[*i*]	Type of each spot along the section *i* (stop or intersection/turn)	dim	
CongestionRate	Relative level of traffic congestion comparing that of year 1998	dim	
cycle	Cycle length at each signalized intersection	s	
CriticalGap	Critical Gap at each unsignalized intersection or turn	second	
d1	Uniform delay at each signalized intersection	s	
d2	Overflow delay at each signalized intersection	s	
Delay[*i*]	Estimated Delay time at intersection *i*	s	
dist[*i*,*j*]	Distance of alternative route *j* on section *i*	meters	
distance	Distance of each section	meters	
dwelltime	Present delayed time at each intersection or turn	s	
diameter	Diameter of the tractive wheels	meters	
dummy	Dummy value	dim	
efficiency	Efficiency of the Bus engine system	dim	
EnergyConsumed	Present fuel consumed for each section	*ml*/s	
Energy	Fuel consumed at each time interval	*ml*/s	
fleetsize	Estimated Fleet size of BT serving route 1 for each study year	buses	
grade	Grade profile between spots *i*–1 and *i*	dim	
green	Green time of each signalized intersection	s	

Variable	Description	Unit	Value[1]
GravityTerm[*i*]	Bus gravity factor between spots *i*–1 and *i*	newton	
growthrate	Yearly growth rate of traffic demand	dim	
hp[*i*]	Power output of Bus engine in *i*th gear	hp	
horizonyr	The horizon year of the bus operation simulation analysis	year	2020
IniATP	probability to select auto in 1998	dim	
inidemand	Total Traffic demand of the first study year	persons	
iniBTdemand	BT passenger demand in 1998	persons	
inifleetsize	BT fleet size to meet peak hr.(6:30am-9:30am) passenger demand in 1998	buses	
Intsectype	Classification of the intersection types (Refer appendix ****)	dim	
invehtime[*i*,*j*]	In-vehicle travel time of bus for section *j*, in year *i*	s	
J	differential gearing ratio	dim	4.5
LOS	Level of traffic condition for each alternative route	dim	
MinStopTime	Minimum time lost at each bus stop	s	
mktpnt	Market penetration of AVL user interface in each study year	dim	
nextstpndly	Estimated lose time at next stop or intersection/turn	s	
NumOfRt	Number of sections in study route	dim	7
NumOfAltn	Number of alternative routes in each section	dim	3
NumOperation	Number of round trips during peak hour	dim	5
numyear	Number of study year from 1998, the starting year	dim	
origin	The position from which bus starts such as time-check, stop, intersection/turn	dim	
OTLOS	level of service of Other modes(walk, bycycle)	dim	
OTP	probability to select other modes	dim	
OTRELT	relative time (in-vehicle time / total journey time) of other modes	dim	
OTU	Utility of Other modes	dim	
pcu	Passenger car equivalent unit (=PCE)	dim	
PF	Progression factor	dim	
PushPedal	Intensity of drivers' pushing acceleration pedal	dim	
pax[*i*]	Number of Present passengers on board between stops *i* and *i*+1	persons	
paxMassNewton[*i*]	Total weight of passengers on board between stops *i* and *i*+1	newton	
previousstpndly	Estimated lose time at previous stop or intersection/turn	s	
previouspax	Number of passengers on board between bus stops *i*–1 and *i*	persons	
previousboarding	number of passengers boarding at previous bus stop	persons	
previousalight	number of passengers alighting at previous bus stop	persons	
Qtotal[*i*]	Total traffic volume on section *i* via all alternative routes	*Veh/hr.*	
route	Number of sections in study route	dim	7
RPM[*i*]	Engine Speed for the *i*th gear	RPM	
Rtcap[*i*]	Calibrated capacity of each section *i*	*Veh/hr.*	
Sigmainvtt[*i*]	Sum of times travelling all alternative routes in section *i*	min.	
spot[*i*,1]	X-axis coordinate value of the *i*th spot along the each section	meters	
spot[*i*,2]	Y-axis coordinate value of the *i*th spot along the each section	meters	
spotnum	Number of total spots on each section	dim	
schdltime	Scheduled bus operation time of each section	dim	
stopdelay	Number of intersections and turns along the study section	dim	
stopnumb	Number of total bus stops on each section	dim	
stopdistnt[*i*]	Distance between spots *i*–1 and *i*	meters	
spotheight[*i*]	Geographical height of spot *i*	meters	
stopheight[*i*]	Geographical height of bus stop *i*	meters	

Variable	Description	Unit	Value[1]
stopid[*i*]	Consecutive stop ID for *i*th stop	dim	
StopTime[*i*]	Dwell time at *i*th stop	s	
spots	Number of spots between two stops or intersections/turns	dim	
StopDistance	Stop distance needed to stop at next bus stop or intersection/turn	meters	
TE[*i*,*j*]	Tractive effort of bus engine system for the *j*th gear, engine speed *i*	newton	
time	The indicator representing lapse of bus operation time	s	
Tn[*i*,*j*]	Revised Travel time of section *i*, alternative route *j* at each iteration	min.	
Tnini	First revised value of travel time for each section and alternative route	min.	
To[*i*,*j*]	Free flow travel time using alternative route *j*, section *i*	min.	
totaldist[*i*]	Total distance between origin and present spot	meters	
TotalMassNewton[*i*]	Total weight of bus including passenger on-board between stops *i* and *i*+1	newton	
Totalposition	Present bus travel distance from the beginning of the study route	meters	
Totalenergyco[*i*]	Fuel consumed for entire route in study year *i*	*ml*	
totalpax	Number of total passengers through the study route in each year	persons	
TotalResistance	Total amount of resistance to bus tractive power	newton	
totalTn	Sum of revised travel times for each alternative route at every iteration	min.	
totinvehtime[*i*]	In-vehicle travel time of bus for entire route in year *i*	s	
TractiveEffort	Present tractive effort of the bus engine system	newton	
trafficdemand	Total traffic demand using all the modes(auto, bus, walk, bicycle)	persons	
TGear[*i*]	Tractive effort at *i*th gear	newton	
u[*i*]	Transmission gearing ratio for the *i*th gear	dim	
V[*i*,*j*]	Bus speed in *j*th gear at engine speed *i*	m/s	
Vol[*i*,*j*]	Traffic volume of alternative route *j*, section *i*	*Veh/hr.*	
volume	Traffic volume at each intersection/turn	*Veh/hr.*	
X[*i*]	V/C at *i*th intersection/turn	dim	
year	Study years (from 1998 till 2020)	dim	1998 - 2020

[1]:Values applied to the BT operation simulation model

D.4 BT Operation Simulation Model Output (Section 1; Scenario 1).

```
ROUTE 1
  TIME //BUS VELOCITY //BUS POSITION //ENERGY CONSUMPTION
  (sec)//  (m/sec2)   //    (m)      //      (g)
        SPOT[ 1] IS UNSIGNALIZED RIGHT TURN,   2002
        INTERSECTION DELAY =   30.493
           1      0.000         0.000         1.867
           2      0.000         0.000         3.733
           3      0.000         0.000         5.600
           4      1.673         1.673         7.467
           5      3.345         5.018         9.333
           6      5.017        10.035        12.354
           7      6.517        16.552        15.375
           8      8.544        25.096        19.275
           9     10.376        35.472        23.175
          10     12.114        47.587        27.075
          11     13.717        61.304        32.575
          12     12.717        74.022        38.075
          13     11.717        85.739        41.975
          14     10.717        96.456        45.875
          15      9.717       106.174        49.775
          16      8.717       114.891        53.675
          17      7.717       122.609        56.696
        SPOT[ 4] IS STOP
        STOP DEALY =    7.100
          18      0.000       122.609        58.562
          19      0.000       122.609        60.429
          20      0.000       122.609        62.296
          21      0.000       122.609        64.162
          22      0.000       122.609        66.029
          23      0.000       122.609        67.896
          24      0.000       122.609        69.762
          25      0.000       122.609        71.629
          26      0.000       122.609        73.496
          27      0.000       122.609        75.362
          28      0.000       122.609        77.229
          29      0.000       122.609        79.096
          30      0.000       122.609        80.962
          31      0.000       122.609        82.829
          32      0.000       122.609        84.696
          33      0.000       122.609        86.562
          34      0.000       122.609        88.429
          35      0.000       122.609        90.296
          36      0.000       122.609        92.162
          37      0.000       122.609        94.029
          38      0.000       122.609        95.896
          39      0.000       122.609        97.762
          40      0.000       122.609        99.629
          41      0.000       122.609       101.496
          42      0.000       122.609       103.362
          43      0.000       122.609       105.229
          44      0.000       122.609       107.096
          45      0.000       122.609       108.962
          46      0.000       122.609       110.829
          47      0.000       122.609       112.696
          48      0.000       122.609       114.562
          49      1.528       124.137       116.429
          50      3.253       127.389       118.296
          51      4.976       132.366       121.317
          52      6.543       138.909       124.337
          53      8.634       147.543       128.237
          54     10.522       158.065       132.137
          55     12.311       170.376       136.037
          56     11.311       181.686       139.937
          57     10.311       191.997       143.837
          58      9.311       201.307       147.737
          59      8.311       209.618       150.758
        SPOT[ 9] IS STOP
        STOP DEALY =    7.100
          60      0.000       209.618       152.625
          61      0.000       209.618       154.492
          62      0.000       209.618       156.358
          63      0.000       209.618       158.225
          64      0.000       209.618       160.092
          65      0.000       209.618       161.958
          66      0.000       209.618       163.825
          67      0.000       209.618       165.692
          68      1.541       211.158       167.558
          69      3.138       214.297       169.425
          70      4.735       219.032       172.446
          71      6.172       225.204       175.467
          72      8.152       233.356       178.487
          73      9.944       243.300       182.387
          74     11.564       254.864       186.287
          75     13.119       267.983       191.787
          76     14.553       282.536       197.287
          77     15.995       298.531       202.787
          78     14.995       313.527       208.287
          79     13.995       327.522       213.787
          80     12.995       340.517       219.287
          81     11.995       352.512       223.187
          82     10.995       363.507       227.087
          83      9.995       373.502       230.988
          84      8.995       382.497       234.888
        SPOT[16] IS STOP
        STOP DEALY =   10.100
          85      0.000       382.497       236.754
          86      0.000       382.497       238.621
          87      0.000       382.497       240.488
          88      0.000       382.497       242.354
          89      0.000       382.497       244.221
          90      0.000       382.497       246.088
          91      0.000       382.497       247.954
          92      0.000       382.497       249.821
          93      1.367       383.864       251.688
          94      3.111       386.975       253.554
          95      4.855       391.830       256.575
          96      6.458       398.288       259.596
          97      8.585       406.873       263.496
          98     10.508       417.380       267.396
          99     12.321       429.701       271.296
         100     13.994       443.695       276.796
         101     15.538       459.234       282.296
         102     17.089       476.322       288.429
         103     18.551       494.873       294.562
         104     19.869       514.742       300.696
         105     21.127       535.869       306.829
         106     20.127       555.995       312.962
         107     19.127       575.122       319.096
         108     18.127       593.249       325.229
         109     17.127       610.376       331.362
         110     16.127       626.502       336.862
         111     15.127       641.629       342.362
         112     14.127       655.756       347.862
         113     13.127       668.882       353.362
         114     12.127       681.009       357.262
         115     11.127       692.136       361.162
         116     10.127       702.263       365.062
         117      9.127       711.389       368.962
         118      8.127       719.516       371.983
         119      7.127       726.643       375.004
        SPOT[18] IS SIGNALIZED RIGHT TURN,     1002
        INTERSECTION DELAY =   30.515
         120      0.000       726.643       376.871
         121      0.000       726.643       378.737
         122      0.000       726.643       380.604
         123      0.000       726.643       382.471
         124      0.000       726.643       384.337
         125      0.000       726.643       386.204
         126      0.000       726.643       388.071
         127      0.000       726.643       389.937
         128      0.000       726.643       391.804
         129      0.000       726.643       393.671
         130      0.000       726.643       395.537
         131      1.423       728.066       397.404
         132      3.048       731.113       399.271
```

```
 133      4.671        735.785        402.292
 134      6.148        741.933        405.312
 135      8.163        750.096        408.333
 136      9.987        760.084        412.233
 137      8.987        769.071        416.133
 138      7.987        777.059        419.154
 139      6.987        784.046        422.175
 140      5.987        790.033        425.196
 141      4.987        795.021        428.217
SPOT[32] IS SIGNALIZED LEFT TURN,      1003
INTERSECTION DELAY =   22.750
 142      0.000        795.021        430.083
 143      0.000        795.021        431.950
 144      0.000        795.021        433.817
 145      0.000        795.021        435.683
 146      0.000        795.021        437.550
 147      0.000        795.021        439.417
 148      0.000        795.021        441.283
 149      0.000        795.021        443.150
 150      0.000        795.021        445.017
 151      0.000        795.021        446.883
 152      0.000        795.021        448.750
 153      0.000        795.021        450.617
 154      0.000        795.021        452.483
 155      0.000        795.021        454.350
 156      0.000        795.021        456.217
 157      0.000        795.021        458.083
 158      0.000        795.021        459.950
 159      0.000        795.021        461.817
 160      0.000        795.021        463.683
 161      0.000        795.021        465.550
 162      0.000        795.021        467.417
 163      0.000        795.021        469.283
 164      0.000        795.021        471.150
 165      0.000        795.021        473.017
 166      0.000        795.021        474.883
 167      0.000        795.021        476.750
 168      0.000        795.021        478.617
 169      0.000        795.021        480.483
 170      0.000        795.021        482.350
 171      0.000        795.021        484.217
 172      0.000        795.021        486.083
 173      1.554        796.575        487.950
 174      3.195        799.770        489.817
 175      4.835        804.606        492.837
 176      6.316        810.922        495.858
 177      8.330        819.251        498.879
 178     10.151        829.403        502.779
 179     11.798        841.201        506.679
 180     13.384        854.585        512.179
 181     14.849        869.435        517.679
 182     16.324        885.758        523.179
 183     17.715        903.473        529.312
 184     19.029        922.502        535.446
 185     20.213        942.716        541.579
 186     21.344        964.060        547.712
 187     22.424        986.484        553.846
 188     23.460       1009.944        559.979
 189     22.460       1032.403        566.112
 190     21.460       1053.863        572.246
 191     20.460       1074.322        578.379
 192     19.460       1093.782        584.512
 193     18.460       1112.241        590.646
 194     17.460       1129.701        596.779
 195     16.460       1146.160        602.279
 196     15.460       1161.620        607.779
 197     14.460       1176.079        613.279
 198     13.460       1189.539        618.779
 199     12.460       1201.998        622.679
 200     11.460       1213.458        626.579
 201     10.460       1223.917        630.479
SPOT[38] IS SIGNALIZED STRIGHT TURN,   1001
INTERSECTION DELAY =   10.037
 202      0.000       1223.917        632.346
 203      0.000       1223.917        634.212
 204      0.000       1223.917        636.079
 205      0.000       1223.917        637.946
 206      0.000       1223.917        639.812
 207      0.000       1223.917        641.679
```

```
 208      0.000       1223.917        643.546
 209      0.000       1223.917        645.412
 210      0.000       1223.917        647.279
 211      0.000       1223.917        649.146
 212      0.000       1223.917        651.012
 213      0.000       1223.917        652.879
 214      0.000       1223.917        654.746
 215      0.000       1223.917        656.612
 216      0.000       1223.917        658.479
 217      0.000       1223.917        660.346
 218      0.000       1223.917        662.212
 219      0.000       1223.917        664.079
 220      0.000       1223.917        665.946
 221      0.000       1223.917        667.812
 222      0.000       1223.917        669.679
 223      0.000       1223.917        671.546
 224      0.000       1223.917        673.412
 225      1.314       1225.231        675.279
 226      3.006       1228.237        677.146
 227      4.697       1232.934        680.167
 228      6.253       1239.187        683.187
 229      8.340       1247.527        687.087
 230     10.229       1257.756        690.987
 231     12.003       1269.759        694.887
 232     13.639       1283.399        700.387
 233     15.150       1298.549        705.887
 234     16.664       1315.213        711.387
 235     18.093       1333.306        717.521
 236     19.443       1352.749        723.654
 237     20.665       1373.413        729.787
 238     19.665       1393.078        735.921
 239     18.665       1411.743        742.054
 240     17.665       1429.407        748.187
 241     16.665       1446.072        753.687
 242     15.665       1461.736        759.187
 243     14.665       1476.401        764.687
 244     13.665       1490.065        770.187
 245     12.665       1502.730        775.687
 246     11.665       1514.394        779.587
 247     10.665       1525.059        783.487
SPOT[39] IS STOP
STOP DEALY =    7.100
 248      0.000       1525.059        785.354
 249      0.000       1525.059        787.221
 250      0.000       1525.059        789.087
 251      0.000       1525.059        790.954
 252      0.000       1525.059        792.821
 253      0.000       1525.059        794.687
 254      0.000       1525.059        796.554
 255      0.000       1525.059        798.421
 256      0.000       1525.059        800.287
 257      0.000       1525.059        802.154
 258      0.000       1525.059        804.021
 259      1.351       1526.410        805.887
 260      3.013       1529.423        807.754
 261      4.675       1534.098        810.775
 262      6.195       1540.293        813.796
 263      8.256       1548.549        816.817
 264     10.120       1558.670        820.717
 265     11.803       1570.473        824.617
 266     10.803       1581.276        828.517
 267      9.803       1591.079        832.417
 268      8.803       1599.882        836.317
SPOT[42] IS STOP
STOP DEALY =    7.100
 269      0.000       1599.882        838.183
 270      0.000       1599.882        840.050
 271      0.000       1599.882        841.917
 272      0.000       1599.882        843.783
 273      0.000       1599.882        845.650
 274      0.000       1599.882        847.517
 275      0.000       1599.882        849.383
 276      0.000       1599.882        851.250
 277      1.442       1601.324        853.117
 278      3.093       1604.417        854.983
 279      4.744       1609.161        858.004
 280      6.245       1615.406        861.025
 281      8.283       1623.689        864.046
 282     10.127       1633.816        867.946
 283     11.793       1645.608        871.846
 284     13.391       1659.000        877.346
 285     14.867       1673.867        882.846
 286     13.867       1687.734        888.346
 287     12.867       1700.601        893.846
 288     11.867       1712.468        897.746
 289     10.867       1723.335        901.646
 290      9.867       1733.202        905.546
 291      8.867       1742.069        909.446
SPOT[44] IS UNSIGNALIZED RIGHT TURN,   2002
INTERSECTION DELAY =    3.000
 292      0.000       1742.069        911.312
 293      0.000       1742.069        913.179
 294      0.000       1742.069        915.046
 295      0.000       1742.069        916.912
 296      0.000       1742.069        918.779
 297      0.000       1742.069        920.646
 298      0.000       1742.069        922.512
 299      0.000       1742.069        924.379
 300      1.429       1743.497        926.246
 301      3.072       1746.569        928.113
 302      4.714       1751.283        931.133
 303      6.208       1757.491        934.154
 304      8.242       1765.733        937.175
 305     10.081       1775.814        941.075
 306      9.081       1784.895        944.975
 307      8.081       1792.975        947.996
 308      7.081       1800.056        951.017
 309      6.081       1806.137        954.038
SPOT[45] IS STOP
STOP DEALY =    7.100
 310      0.000       1806.137        955.904
 311      0.000       1806.137        957.771
 312      0.000       1806.137        959.638
 313      0.000       1806.137        961.504
 314      1.443       1807.580        963.371
 315      3.129       1810.710        965.238
 316      4.814       1815.524        968.258
 317      6.349       1821.873        971.279
 318      8.420       1830.293        975.179
 319     10.292       1840.585        979.079
 320     12.055       1852.640        982.979
 321     13.679       1866.319        988.479
 322     15.177       1881.496        993.979
 323     14.177       1895.673        999.479
 324     13.177       1908.850       1004.979
 325     12.177       1921.027       1008.879
 326     11.177       1932.204       1012.779
 327     10.177       1942.381       1016.679
SPOT[47] IS STOP
STOP DEALY =    7.100
 328      0.000       1942.381       1018.546
 329      0.000       1942.381       1020.413
 330      0.000       1942.381       1022.279
 331      0.000       1942.381       1024.146
 332      0.000       1942.381       1026.013
 333      0.000       1942.381       1027.879
 334      0.000       1942.381       1029.746
 335      0.000       1942.381       1031.613
 336      1.374       1943.754       1033.479
 337      3.085       1946.839       1035.346
 338      4.795       1951.634       1038.367
 339      6.361       1957.996       1041.387
 340      8.466       1966.462       1045.287
 341     10.366       1976.828       1049.188
 342     12.155       1988.983       1053.088
 343     13.800       2002.784       1058.588
 344     12.800       2015.584       1064.088
 345     11.800       2027.385       1067.988
 346     10.800       2038.185       1071.888
 347      9.800       2047.986       1075.788
 348      8.800       2056.786       1079.688
 349      7.800       2064.587       1082.708
SPOT[48] IS UNSIGNALIZED LEFT TURN,    2003
INTERSECTION DELAY =   10.751
 350      0.000       2064.587       1084.575
 351      0.000       2064.587       1086.442
 352      0.000       2064.587       1088.308
 353      0.000       2064.587       1090.175
```

```
 354      0.000       2064.587       1092.042
 355      0.000       2064.587       1093.908
 356      0.000       2064.587       1095.775
 357      0.000       2064.587       1097.642
 358      0.000       2064.587       1099.508
 359      1.670       2066.257       1101.375
 360      3.340       2069.597       1103.242
 361      5.009       2074.607       1106.262
 362      6.503       2081.110       1109.283
 363      8.537       2089.647       1113.183
 364      7.537       2097.183       1116.204
SPOT[54] IS STOP
STOP DEALY =   25.100
 365      0.000       2097.183       1118.071
 366      0.000       2097.183       1119.937
 367      0.000       2097.183       1121.804
 368      0.000       2097.183       1123.671
 369      0.000       2097.183       1125.537
 370      0.000       2097.183       1127.404
 371      0.000       2097.183       1129.271
 372      0.000       2097.183       1131.137
 373      0.000       2097.183       1133.004
 374      0.000       2097.183       1134.871
 375      0.000       2097.183       1136.737
 376      1.738       2098.921       1138.604
 377      3.419       2102.340       1140.471
 378      5.100       2107.440       1143.492
 379      6.599       2114.039       1146.512
 380      8.632       2122.670       1150.412
 381     10.465       2133.135       1154.312
 382     12.207       2145.342       1158.212
 383     13.812       2159.154       1163.712
 384     15.294       2174.448       1169.212
 385     16.785       2191.232       1175.346
 386     18.190       2209.422       1181.479
 387     17.190       2226.612       1187.612
 388     16.190       2242.803       1193.112
 389     15.190       2257.993       1198.612
 390     14.190       2272.183       1204.112
 391     13.190       2285.373       1209.612
 392     12.190       2297.563       1213.512
 393     11.190       2308.753       1217.412
 394     10.190       2318.944       1221.312
 395      9.190       2328.134       1225.212
 396      8.190       2336.324       1228.233
 397      7.190       2343.514       1231.254
SPOT[59] IS STOP
STOP DEALY =    0.000
 398      0.000       2343.514       1233.121
 399      0.000       2343.514       1234.987
 400      0.000       2343.514       1236.854
 401      0.000       2343.514       1238.721
 402      0.000       2343.514       1240.587
 403      0.000       2343.514       1242.454
 404      0.000       2343.514       1244.321
 405      0.000       2343.514       1246.187
 406      0.000       2343.514       1248.054
 407      0.000       2343.514       1249.921
 408      0.000       2343.514       1251.787
 409      0.000       2343.514       1253.654
 410      0.000       2343.514       1255.521
 411      0.000       2343.514       1257.387
 412      0.000       2343.514       1259.254
 413      0.000       2343.514       1261.121
 414      0.000       2343.514       1262.987
 415      0.000       2343.514       1264.854
 416      0.000       2343.514       1266.721
 417      0.000       2343.514       1268.587
 418      0.000       2343.514       1270.454
 419      0.000       2343.514       1272.321
 420      0.000       2343.514       1274.187
 421      0.000       2343.514       1276.054
 422      0.000       2343.514       1277.921
 423      0.000       2343.514       1279.787
 424      1.348       2344.862       1281.654
 425      3.123       2347.985       1283.521
 426      4.897       2352.883       1286.542
 427      6.523       2359.406       1289.562
 428      8.704       2368.109       1293.462
 429     10.662       2378.772       1297.362
 430     12.506       2391.278       1302.862
 431     14.196       2405.474       1308.362
 432     15.856       2421.330       1313.862
 433     17.414       2438.744       1319.996
 434     16.414       2455.157       1325.496
 435     15.414       2470.571       1330.996
 436     14.414       2484.985       1336.496
 437     13.414       2498.399       1341.996
 438     12.414       2510.813       1345.896
 439     11.414       2522.227       1349.796
 440     10.414       2532.641       1353.696
 441      9.414       2542.055       1357.596
 442      8.414       2550.469       1361.496
 443      7.414       2557.883       1364.517
 444      6.414       2564.297       1367.537
SPOT[62] IS UNSIGNALIZED LEFT TURN,    2003
INTERSECTION DELAY =   57.097
 445      0.000       2564.297       1369.404
 446      1.522       2565.819       1371.271
 447      2.541       2568.360       1373.137
 448      4.255       2572.615       1376.158
 449      5.803       2578.418       1379.179
 450      7.995       2586.413       1382.200
 451      9.974       2596.387       1386.100
 452     11.759       2608.146       1390.000
 453     13.439       2621.585       1395.500
 454     14.979       2636.564       1401.000
 455     16.518       2653.082       1406.500
 456     17.964       2671.045       1412.633
 457     19.324       2690.370       1418.767
 458     20.545       2710.915       1424.900
 459     19.545       2730.460       1431.033
 460     18.545       2749.005       1437.167
 461     17.545       2766.550       1443.300
 462     16.545       2783.095       1448.800
 463     15.545       2798.640       1454.300
 464     14.545       2813.186       1459.800
 465     13.545       2826.731       1465.300
 466     12.545       2839.276       1470.800
 467     11.545       2850.821       1474.700
 468     10.545       2861.366       1478.600
SPOT[63] IS STOP
STOP DEALY =   13.100
 469      0.000       2861.366       1480.467
 470      0.000       2861.366       1482.333
 471      0.000       2861.366       1484.200
 472      0.000       2861.366       1486.067
 473      0.000       2861.366       1487.933
 474      0.000       2861.366       1489.800
 475      0.000       2861.366       1491.667
 476      0.000       2861.366       1493.533
 477      0.000       2861.366       1495.400
 478      0.000       2861.366       1497.267
 479      0.000       2861.366       1499.133
 480      0.000       2861.366       1501.000
 481      0.000       2861.366       1502.867
 482      0.000       2861.366       1504.733
 483      0.000       2861.366       1506.600
 484      0.000       2861.366       1508.467
 485      0.000       2861.366       1510.333
 486      0.000       2861.366       1512.200
 487      0.000       2861.366       1514.067
 488      0.000       2861.366       1515.933
 489      0.000       2861.366       1517.800
 490      0.000       2861.366       1519.667
 491      0.000       2861.366       1521.533
 492      0.000       2861.366       1523.400
 493      0.000       2861.366       1525.267
 494      0.000       2861.366       1527.133
 495      0.000       2861.366       1529.000
 496      0.000       2861.366       1530.867
 497      0.000       2861.366       1532.733
 498      0.000       2861.366       1534.600
 499      0.000       2861.366       1536.467
 500      0.000       2861.366       1538.333
 501      0.000       2861.366       1540.200
 502      0.000       2861.366       1542.067
 503      0.000       2861.366       1543.933
 504      0.000       2861.366       1545.800
 505      0.000       2861.366       1547.667
 506      0.000       2861.366       1549.533
 507      0.000       2861.366       1551.400
 508      0.000       2861.366       1553.267
 509      0.000       2861.366       1555.133
 510      0.000       2861.366       1557.000
 511      0.000       2861.366       1558.867
 512      0.000       2861.366       1560.733
 513      0.000       2861.366       1562.600
 514      0.000       2861.366       1564.467
 515      0.000       2861.366       1566.333
 516      0.000       2861.366       1568.200
 517      0.000       2861.366       1570.067
 518      0.000       2861.366       1571.933
 519      0.000       2861.366       1573.800
 520      0.000       2861.366       1575.667
 521      0.000       2861.366       1577.533
 522      0.000       2861.366       1579.400
 523      0.000       2861.366       1581.267
 524      0.000       2861.366       1583.133
 525      0.000       2861.366       1585.000
 526      0.000       2861.366       1586.867
 527      1.361       2862.727       1588.733
 528      3.120       2865.847       1590.600
 529      4.877       2870.724       1593.621
 530      6.486       2877.210       1596.642
 531      8.651       2885.861       1600.542
 532     10.596       2896.457       1604.442
 533     12.426       2908.883       1608.342
 534     11.426       2920.309       1612.242
 535     10.426       2930.734       1616.142
 536      9.426       2940.160       1620.042
 537      8.426       2948.586       1623.942
 538      7.426       2956.012       1626.962
 539      6.426       2962.438       1629.983
SPOT[64] IS STOP
STOP DEALY =    7.100
 540      0.000       2962.438       1631.850
 541      0.000       2962.438       1633.717
 542      0.000       2962.438       1635.583
 543      0.000       2962.438       1637.450
 544      0.000       2962.438       1639.317
 545      0.000       2962.438       1641.183
 546      0.000       2962.438       1643.050
 547      0.000       2962.438       1644.917
 548      0.000       2962.438       1646.783
 549      0.000       2962.438       1648.650
 550      0.000       2962.438       1650.517
 551      0.000       2962.438       1652.383
 552      0.000       2962.438       1654.250
 553      0.000       2962.438       1656.117
 554      1.504       2963.943       1657.983
 555      3.199       2967.141       1659.850
 556      4.892       2972.033       1662.871
 557      6.424       2978.457       1665.892
 558      8.505       2986.962       1669.792
 559     10.380       2997.343       1673.692
 560     12.149       3009.492       1677.592
 561     13.774       3023.265       1683.092
 562     15.268       3038.533       1688.592
 563     14.268       3052.802       1694.092
 564     13.268       3066.070       1699.592
 565     12.268       3078.338       1703.492
 566     11.268       3089.607       1707.392
 567     10.268       3099.875       1711.292
 568      9.268       3109.143       1715.192
SPOT[65] IS SIGNALIZED RIGHT TURN,     1002
INTERSECTION DELAY =   23.234
 569      0.000       3109.143       1717.058
 570      0.000       3109.143       1718.925
 571      0.000       3109.143       1720.792
 572      0.000       3109.143       1722.658
 573      0.000       3109.143       1724.525
 574      0.000       3109.143       1726.392
 575      0.000       3109.143       1728.258
 576      0.000       3109.143       1730.125
 577      0.000       3109.143       1731.992
 578      1.721       3110.864       1733.858
 579      3.442       3114.306       1735.725
 580      5.162       3119.468       1738.746
 581      6.697       3126.165       1741.767
 582      8.776       3134.941       1745.667
 583      7.776       3142.717       1748.687
 584      6.776       3149.493       1751.708
 585      5.776       3155.269       1754.729
SPOT[68] IS STOP
STOP DEALY =   19.100
 586      0.000       3155.269       1756.596
 587      0.000       3155.269       1758.462
 588      0.000       3155.269       1760.329
 589      0.000       3155.269       1762.196
 590      0.000       3155.269       1764.062
 591      0.000       3155.269       1765.929
 592      0.000       3155.269       1767.796
 593      0.000       3155.269       1769.662
 594      0.000       3155.269       1771.529
 595      0.000       3155.269       1773.396
 596      0.000       3155.269       1775.262
 597      0.000       3155.269       1777.129
 598      0.000       3155.269       1778.996
 599      0.000       3155.269       1780.862
 600      0.000       3155.269       1782.729
 601      0.000       3155.269       1784.596
 602      0.000       3155.269       1786.462
 603      0.000       3155.269       1788.329
 604      0.000       3155.269       1790.196
 605      0.000       3155.269       1792.062
 606      0.000       3155.269       1793.929
 607      0.000       3155.269       1795.796
 608      0.000       3155.269       1797.662
 609      0.000       3155.269       1799.529
 610      1.551       3156.821       1801.396
 611      3.300       3160.121       1803.262
 612      5.048       3165.169       1806.283
 613      6.629       3171.798       1809.304
 614      8.755       3180.553       1813.204
 615     10.667       3191.221       1817.104
 616     12.477       3203.697       1821.004
 617     14.138       3217.835       1826.504
 618     15.666       3233.502       1832.004
 619     17.203       3250.704       1838.137
 620     16.203       3266.907       1843.637
 621     15.203       3282.110       1849.137
 622     14.203       3296.313       1854.637
 623     13.203       3309.516       1860.137
 624     12.203       3321.719       1864.037
 625     11.203       3332.922       1867.937
 626     10.203       3343.124       1871.837
 627      9.203       3352.327       1875.737
SPOT[70] IS STOP
STOP DEALY =    0.000
 628      0.000       3352.327       1877.604
 629      0.000       3352.327       1879.471
 630      0.000       3352.327       1881.337
 631      0.000       3352.327       1883.204
 632      0.000       3352.327       1885.071
 633      0.000       3352.327       1886.937
 634      0.000       3352.327       1888.804
 635      0.000       3352.327       1890.671
 636      0.000       3352.327       1892.537
 637      0.000       3352.327       1894.404
 638      0.000       3352.327       1896.271
 639      0.000       3352.327       1898.137
 640      0.000       3352.327       1900.004
 641      0.000       3352.327       1901.871
 642      0.000       3352.327       1903.737
 643      0.000       3352.327       1905.604
 644      0.000       3352.327       1907.471
 645      0.000       3352.327       1909.337
 646      0.000       3352.327       1911.204
 647      0.000       3352.327       1913.071
 648      1.494       3353.821       1914.937
 649      3.305       3357.125       1916.804
 650      5.114       3362.240       1919.825
 651      6.757       3368.997       1922.846
 652      8.959       3377.956       1926.746
 653     10.930       3388.887       1930.646
```

```
 654     12.795       3401.681       1936.146
 655     14.500       3416.181       1941.646
 656     16.179       3432.360       1947.146
 657     15.179       3447.539       1952.646
 658     14.179       3461.717       1958.146
 659     13.179       3474.896       1963.646
 660     12.179       3487.075       1967.546
 661     11.179       3498.253       1971.446
 662     10.179       3508.432       1975.346
 663      9.179       3517.611       1979.246
 664      8.179       3525.789       1982.267
 665      7.179       3532.968       1985.287
SPOT[72] IS STOP
STOP DEALY =   10.100
 666      0.000       3532.968       1987.154
 667      1.455       3534.423       1989.021
 668      2.403       3536.826       1990.887
 669      4.145       3540.972       1992.754
 670      5.725       3546.697       1995.775
 671      7.199       3553.896       1998.796
 672      9.226       3563.122       2002.696
 673     11.052       3574.174       2006.596
 674     12.816       3586.990       2012.096
 675     14.435       3601.425       2017.596
 676     16.039       3617.464       2023.096
 677     15.039       3632.503       2028.596
 678     14.039       3646.542       2034.096
 679     13.039       3659.581       2039.596
 680     12.039       3671.619       2043.496
 681     11.039       3682.658       2047.396
 682     10.039       3692.697       2051.296
SPOT[84] IS UNSIGNALIZED LEFT TURN,    2003
INTERSECTION DELAY =   25.124
 683      0.000       3692.697       2053.162
 684      0.000       3692.697       2055.029
 685      0.000       3692.697       2056.896
 686      0.000       3692.697       2058.762
 687      0.000       3692.697       2060.629
 688      0.000       3692.697       2062.496
 689      0.000       3692.697       2064.362
 690      0.000       3692.697       2066.229
 691      0.000       3692.697       2068.096
 692      0.000       3692.697       2069.962
 693      0.000       3692.697       2071.829
 694      1.418       3694.116       2073.696
 695      3.237       3697.352       2075.562
 696      5.054       3702.406       2078.583
 697      6.713       3709.119       2081.604
 698      8.930       3718.049       2085.504
 699     10.916       3728.965       2089.404
 700     12.791       3741.756       2094.904
 701     14.506       3756.262       2100.404
 702     16.193       3772.455       2105.904
 703     17.773       3790.228       2112.037
 704     19.256       3809.484       2118.171
 705     20.583       3830.067       2124.304
 706     21.845       3851.912       2130.437
 707     23.045       3874.957       2136.571
 708     24.189       3899.146       2142.704
 709     25.281       3924.428       2148.837
 710     26.323       3950.751       2154.971
 711     27.318       3978.069       2161.104
 712     28.266       4006.335       2167.237
 713     27.266       4033.602       2173.371
 714     26.266       4059.868       2179.504
 715     25.266       4085.135       2185.637
 716     24.266       4109.401       2191.771
 717     23.266       4132.668       2197.904
 718     22.266       4154.934       2204.037
 719     21.266       4176.201       2210.171
 720     20.266       4196.467       2216.304
 721     19.266       4215.733       2222.437
 722     18.266       4234.000       2228.571
 723     17.266       4251.266       2234.704
 724     16.266       4267.533       2240.204
 725     15.266       4282.799       2245.704
 726     14.266       4297.066       2251.204
 727     13.266       4310.332       2256.704
 728     12.266       4322.599       2260.604
 729     11.266       4333.865       2264.504
 730     10.266       4344.132       2268.404
SPOT[85] IS STOP
STOP DEALY =   10.100
 731      0.000       4344.132       2270.271
 732      0.000       4344.132       2272.137
 733      0.000       4344.132       2274.004
 734      0.000       4344.132       2275.871
 735      0.000       4344.132       2277.737
 736      0.000       4344.132       2279.604
 737      0.000       4344.132       2281.471
 738      0.000       4344.132       2283.337
 739      0.000       4344.132       2285.204
 740      0.000       4344.132       2287.071
 741      0.000       4344.132       2288.937
 742      0.000       4344.132       2290.804
 743      0.000       4344.132       2292.671
 744      0.000       4344.132       2294.537
 745      0.000       4344.132       2296.404
 746      0.000       4344.132       2298.271
 747      0.000       4344.132       2300.137
 748      0.000       4344.132       2302.004
 749      0.000       4344.132       2303.871
 750      0.000       4344.132       2305.737
 751      0.000       4344.132       2307.604
 752      0.000       4344.132       2309.471
 753      0.000       4344.132       2311.337
 754      0.000       4344.132       2313.204
 755      0.000       4344.132       2315.071
 756      0.000       4344.132       2316.937
 757      0.000       4344.132       2318.804
 758      1.712       4345.844       2320.671
 759      3.424       4349.268       2322.537
 760      5.135       4354.404       2325.558
 761      6.652       4361.056       2328.579
 762      8.746       4369.802       2332.479
 763      7.746       4377.547       2335.500
 764      6.746       4384.293       2338.521
SPOT[86] IS UNSIGNALIZED LEFT TURN,    2003
INTERSECTION DELAY =   20.265
 765      0.000       4384.293       2340.387
 766      0.000       4384.293       2342.254
 767      0.000       4384.293       2344.121
 768      0.000       4384.293       2345.987
 769      0.000       4384.293       2347.854
 770      0.000       4384.293       2349.721
 771      0.000       4384.293       2351.587
 772      0.000       4384.293       2353.454
 773      0.000       4384.293       2355.321
 774      0.000       4384.293       2357.187
 775      0.000       4384.293       2359.054
 776      1.407       4385.700       2360.921
 777      3.246       4388.947       2362.787
 778      5.084       4394.031       2365.808
 779      6.762       4400.793       2368.829
 780      9.005       4409.798       2372.729
 781     11.010       4420.809       2376.629
 782     12.903       4433.712       2382.129
 783     14.632       4448.344       2387.629
 784     16.332       4464.675       2393.129
 785     17.923       4482.598       2399.262
 786     19.414       4502.012       2405.396
 787     18.414       4520.427       2411.529
 788     17.414       4537.841       2417.662
 789     16.414       4554.255       2423.162
 790     15.414       4569.669       2428.662
 791     14.414       4584.084       2434.162
 792     13.414       4597.498       2439.662
 793     12.414       4609.912       2443.562
 794     11.414       4621.327       2447.462
 795     10.414       4631.741       2451.362
SPOT[88] IS STOP
STOP DEALY =   43.100
 796      0.000       4631.741       2453.229
 797      0.000       4631.741       2455.096
 798      0.000       4631.741       2456.962
 799      0.000       4631.741       2458.829
 800      0.000       4631.741       2460.696
 801      0.000       4631.741       2462.562
 802      0.000       4631.741       2464.429
 803      0.000       4631.741       2466.296
 804      0.000       4631.741       2468.162
 805      0.000       4631.741       2470.029
 806      0.000       4631.741       2471.896
 807      0.000       4631.741       2473.762
 808      0.000       4631.741       2475.629
 809      0.000       4631.741       2477.496
 810      0.000       4631.741       2479.362
 811      0.000       4631.741       2481.229
 812      0.000       4631.741       2483.096
 813      0.000       4631.741       2484.962
 814      0.000       4631.741       2486.829
 815      0.000       4631.741       2488.696
 816      0.000       4631.741       2490.562
 817      1.482       4633.223       2492.429
 818      3.378       4636.601       2494.296
 819      5.273       4641.874       2497.317
 820      7.003       4648.877       2500.337
 821      9.281       4658.158       2504.237
 822     11.317       4669.475       2508.137
 823     10.317       4679.791       2512.037
 824      9.317       4689.108       2515.937
 825      8.317       4697.424       2518.958
 826      7.317       4704.741       2521.979
 827      6.317       4711.058       2525.000
 828      5.317       4716.374       2528.021
 829      0.000       4716.374       2529.887
 830      0.000       4716.374       2531.754
 831      0.000       4716.374       2533.621
 832      0.000       4716.374       2535.487
 833      0.000       4716.374       2537.354
 834      0.000       4716.374       2539.221
 835      0.000       4716.374       2541.087
 836      0.000       4716.374       2542.954
 837      0.000       4716.374       2544.821
 838      0.000       4716.374       2546.687
 839      0.000       4716.374       2548.554
 840      0.000       4716.374       2550.421
 841      0.000       4716.374       2552.287
 842      0.000       4716.374       2554.154
 843      0.000       4716.374       2556.021
 844      0.000       4716.374       2557.887
 845      0.000       4716.374       2559.754
 846      0.000       4716.374       2561.621
 847      0.000       4716.374       2563.487
 848      0.000       4716.374       2565.354
 849      0.000       4716.374       2567.221
 850      0.000       4716.374       2569.087
 851      0.000       4716.374       2570.954
 852      0.000       4716.374       2572.821
 853      0.000       4716.374       2574.687
 854      0.000       4716.374       2576.554
 855      0.000       4716.374       2578.421
 856      0.000       4716.374       2580.287
 857      0.000       4716.374       2582.154
 858      0.000       4716.374       2584.021
 859      0.000       4716.374       2585.887
 860      0.000       4716.374       2587.754
 861      0.000       4716.374       2589.621
 862      0.000       4716.374       2591.487
 863      0.000       4716.374       2593.354
 864      0.000       4716.374       2595.221
 865      0.000       4716.374       2597.087
 866      0.000       4716.374       2598.954
 867      0.000       4716.374       2600.821
 868      0.000       4716.374       2602.687
 869      0.000       4716.374       2604.554
 870      0.000       4716.374       2606.421
 871      0.000       4716.374       2608.287
 872      0.000       4716.374       2610.154
 873      0.000       4716.374       2612.021
 874      0.000       4716.374       2613.887
 875      0.000       4716.374       2615.754
 876      0.000       4716.374       2617.621
 877      0.000       4716.374       2619.487
 878      0.000       4716.374       2621.354
 879      0.000       4716.374       2623.221
 880      0.000       4716.374       2625.087
```

881	0.000	4716.374	2626.954
882	0.000	4716.374	2628.821
883	0.000	4716.374	2630.687
884	0.000	4716.374	2632.554
885	0.000	4716.374	2634.421
886	0.000	4716.374	2636.287
887	0.000	4716.374	2638.154
888	0.000	4716.374	2640.021
889	0.000	4716.374	2641.888
890	0.000	4716.374	2643.754
891	0.000	4716.374	2645.621
892	0.000	4716.374	2647.488
893	0.000	4716.374	2649.354
894	0.000	4716.374	2651.221
895	0.000	4716.374	2653.088
896	0.000	4716.374	2654.954
897	0.000	4716.374	2656.821
898	0.000	4716.374	2658.688
899	0.000	4716.374	2660.554
900	0.000	4716.374	2662.421

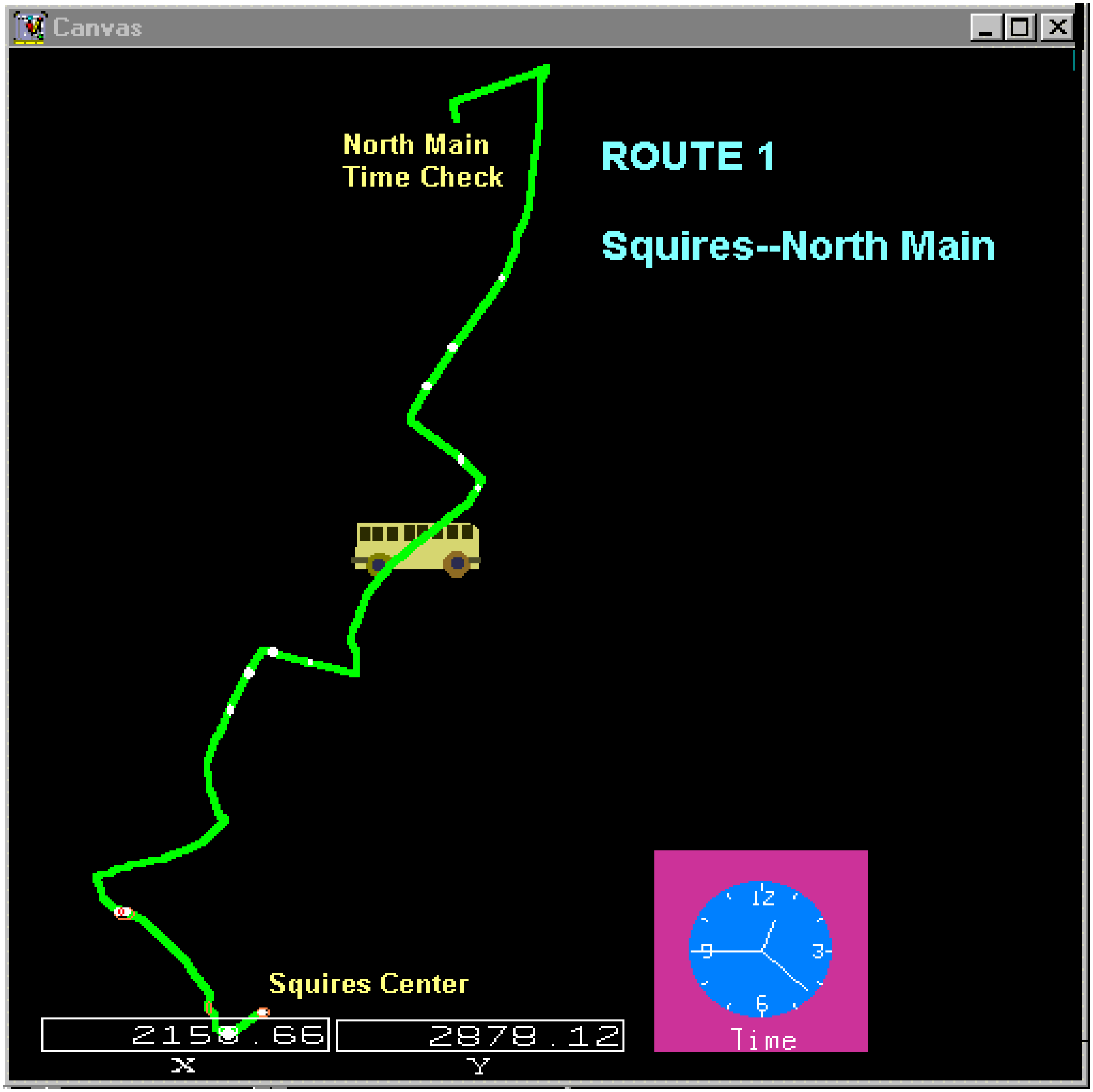

Figure D.1 Graphical Presentation of Bus Simulation Model.