Equilibrated Variable Demand Modelling

Miles Logie
Minnerva Ltd

Structure of the Presentation

- An overview of Variable Demand Modelling topics and issues
- An outline of approaches to obtaining convergent, equilibrium solutions
- An insight into the modelling methodology based on an implementation in OmniTRANS

9th February 2007 Equilibrated Variable Demand Modelling

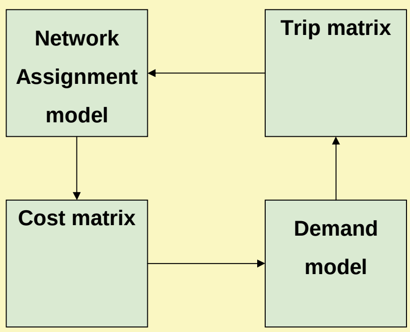
Variable Demand Modelling

- Historically, once demand has been calculated for a forecast year it remains fixed
  - Done using population and 'land use' information, and base year travel costs
  - Modelling then concentrates on the effects on the transport system
- However, changed demand implies costs altered from Base year, so
  - Modelling assumptions are suspect
  - Highway demand levels are easily over-estimated when congestion is high, etc
- Variable Demand Modelling addresses this issue

9th February 2007 Equilibrated Variable Demand Modelling

Variable Demand Modelling

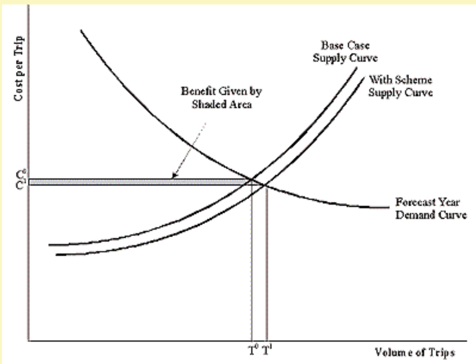
- Resulting (variable demand) model inherently iterative in nature
- Problem is how to stop (at a stable solution)



9th February 2007 Equilibrated Variable Demand Modelling

Issues of Stability

- Transport benefits are the difference between two large quantities for supply and demand
  - Precision is crucial



9th February 2007 Equilibrated Variable Demand Modelling

About Variable Demand Modelling

- DfT has issued guidance on a transport modelling methodology known as Variable Demand Modelling (VDM)

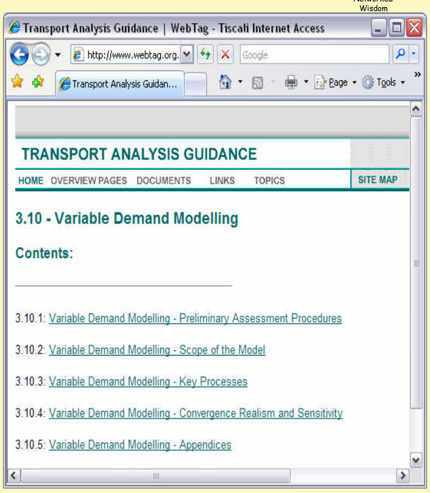
- Objectives
  - Make modelling more realistic for congested conditions
    - Incorporate fuller set of behavioural responses
      - Generation and suppression of trips
      - Decisions on times of travel
    - Improve reliability of scheme and policy assessments
      - Engender stable and robust solutions

- Aspiration to avoid making models unnecessarily complex

9th February 2007 Equilibrated Variable Demand Modelling

DfT VDM Guidance

- Guidance issued June 2006
- Variable Demand Modelling documents on DfT web site
  - http://www.webtag.org.uk/
- Relevant where scale of scheme (> £5m) or level of congestion changes imply scope for significant changes to demand



9th February 2007 Equilibrated Variable Demand Modelling

VDM: Trip Generation – two elements

- Latent Demand ('Trip Generation')
  - Function of demographic, land use, and economic factors
  - Arises as a consequence of the activities that people want to undertake and which require travel
  - Modelled via trip rates
    - DfT TEMPRO program (& NTEM) provides a starting point
- Elastic Demand ('Trip Frequency')
  - Reflects effects of transport system characteristics on decision whether or not to travel, or use a different mode given the costs of travel
  - Modifies Latent demand to produce Expressed demand
  - Modelled via logit choice or elasticity functions

9th February 2007 Equilibrated Variable Demand Modelling

Choice Modelling

- Elasticity models not precluded by VDM advice, but preference expressed for logit models
  - Demand modelling becomes largely a matter of extended hierarchic nested choice logit modelling
- Choice models driven by Composite costs derived from network skim costs
- New VDM guidance provides advice on choice model scaling parameters (λ)

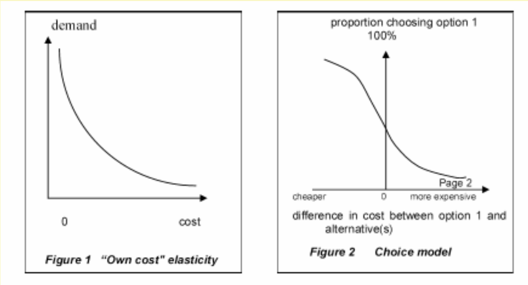
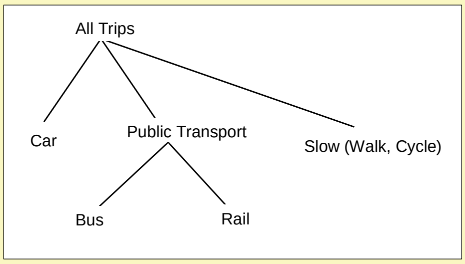


Figure 1 "Own cost" elasticity

Figure 2 Choice model

9th February 2007 Equilibrated Variable Demand Modelling

Choice Hierarchy



- For example, a variable demand model might:
  - first estimate the number of trips from any given origin (trip frequency - often as an elasticity formulation)
  - then estimate how many trips will choose each available mode (mode split); and
  - then estimate how these trips choose amongst the available destinations (trip distribution)

  (Note: this example excludes any time of day choice mechanism.)

9th February 2007 Equilibrated Variable Demand Modelling

Formulations: Trip Distribution

- The general form for a *doubly-constrained distribution model* is:
  - $T_{ij} = a_i * b_j * O_i * D_j * f(G_{ij})$
- Where
  - T_{ij} is the number of trips from zone i to zone j,
  - O_i is the total number of trips originating in zone i
  - D_j is the total number of trips ending in zone j
- The deterrence function f(Gij), in most models is a *logit function*
  - $\exp(-\lambda_{dist}\ G_{ijcomp}) / \{\sum_k \exp(-\lambda_{dist}\ G_{ikcomp})\}$,
- Where
  - G_{ijcomp} is a *composite cost* calculated across the available modes and time periods
- *If these choices are to be calculated after distribution*
  - a_i and b_j are balancing factors which are only used when the model is singly or doubly constrained so that $\sum_j T_{ij} = O_i$ (ie there are O_i trips originating in zone i), and $\sum_i T_{ij} = D_j$ (ie there are D_j trips ending in zone j)
- These are calculated at each iteration of the constraining routine as
  - $a_i = 1/\sum_j b_j D_j f(G_{ij})$ or,
  - $b_j = 1/\sum_i a_i O_i f(G_{ij})$

9th February 2007 Equilibrated Variable Demand Modelling

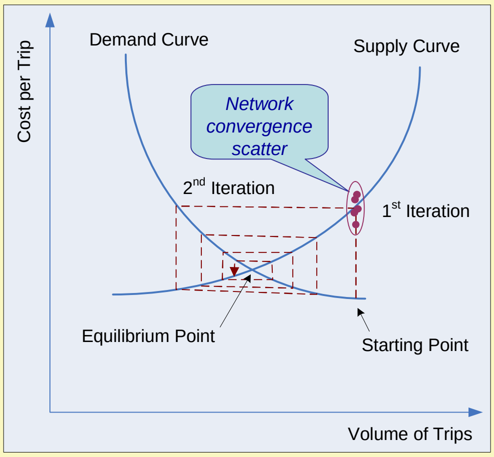
Choice Hierarchy

- For many studies, trip distribution is most important choice, mode choice, time of day choice, and frequency effects may be less significant for particular studies
  - No one VDM formulation applies
- For travellers, choice of mode seems to be most important!

9th February 2007 Equilibrated Variable Demand Modelling

Finding Stable Solutions for Demand *and* Supply

- A simple approach to identifying equilibrium is the ‘cobweb’ approach, but this is not stable.
- Two other approaches:
  - Heuristic, approximate (MSA, Average demand)
  - Mathematical optimisation
    - Problem formulation (objective function for equilibrium – and constraints)
    - Solution method



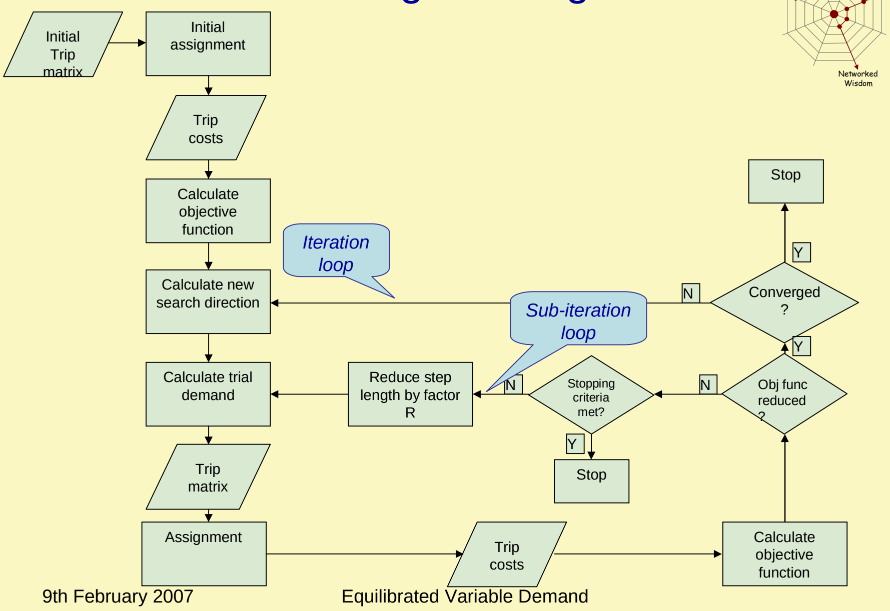
9th February 2007 Equilibrated Variable Demand Modelling

VDM Convergence

- TAG Unit 3.10.4
  - Introduces DIADEM (software component linked to CONTRAM and SATURN)
    - Commissioned by DfT
  - Explains approaches for other software

9th February 2007 Equilibrated Variable Demand Modelling

DIADEM Convergence Algorithm



9th February 2007 Equilibrated Variable Demand Modelling

Equilibrium Algorithms

- Optimising solution based on current and auxiliary solutions (old and latest)
- MSA – Method of Successive Averages
  - Adjusts current and auxiliary based on pre-determined proportions
    - Simple and effective, but does not find true convergence
- Other methods
  - ‘Algorithm 1’
  - DMM – Direct Monotone Method
  - Two-Direction Method
- Standard approach
    - Define objective function
    - Calculate search direction

9th February 2007 Equilibrated Variable Demand Modelling

Equilibration Methods

- General features
  - Objective Function – signals whether equilibrium is being approached
  - Search Direction – calculation of best direction to change existing values
  - Step Length – how much to change values
- Direct Monotone Method
  - Developed by Mike Smith (University of York)
  - Characteristic Objective Function and Search Direction specification

Some Objective Functions

- Taken from DIADEM developments
- Algorithm 1

$$V(X)=\frac{1}{2}\sum_s\{D_s(C(X))-X_s\}^2$$

where:
X = current flows
C(X) = costs based on flows from assignment
D = demand
λ = auxiliary cost estimate
s = segment (commodity)

- Direct Monotone Method (DMM):

$$V(\lambda, X)=\sum_s\left\{X_s^2[C_s(X)-\lambda_s]_+^2+(UX_s-X_s)^2[\lambda_s-C_s(X)]_+^2+\lambda_s^2[X_s-D_s(\lambda)]_+^2+(UC_s-\lambda_s)^2[D_s(\lambda)-X_s]_+^2\right\}$$

VDM Implementation with OmniTRANS

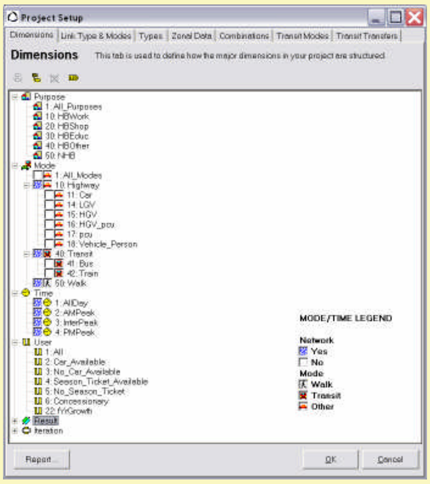
- Facilitates development of VDM (demand) modelling
  - No one approach suited for all studies
- Implements DIADEM methodology
  - Uses DMM solution method
  - Calculates an 'active' set of routes
- Provides integrated, multi-modal solution
  - Adaptable and efficient model development and operation
- Current implementation applied on limited scale only
  - OK, but some further needed optimisation for large scale application
  - PA to OD transformations not in current version
    - Note Jacobs Consultancy work and DfT/DIADEM activity
- Offers an insight into processes

9th February 2007 Equilibrated Variable Demand Modelling

Preliminaries

- Segmentation
  - Purpose
  - Mode
  - Time
  - User

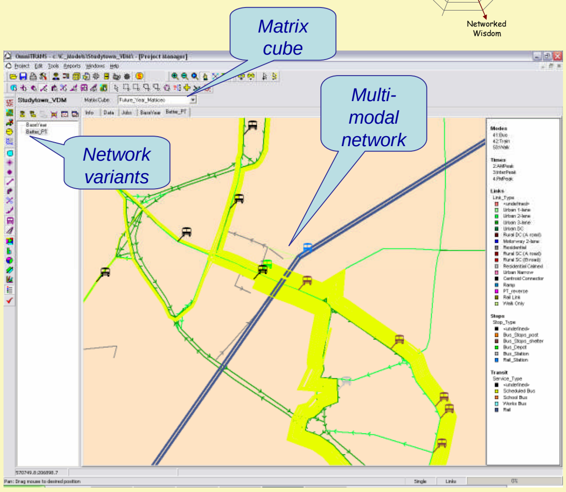
  ('pmtu')
- (RUC/TIF add to demands on segmentation, e.g. income)



9th February 2007 Equilibrated Variable Demand Modelling

Preliminaries

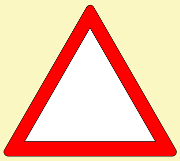
- Base, (Reference), and Future Years:
  - Networks (multi-modal)
  - Trip matrices and zonal data, in matrix cubes
- ('Matrix cubes' package and supply information by pmtu dimensions)



Preliminaries

- Base year matrices (for incremental modelling elements)
- Future year trip ends
  - Synthesis and survey data processing can all be within OmniTRANS, or done externally

An Insight into the Implementation

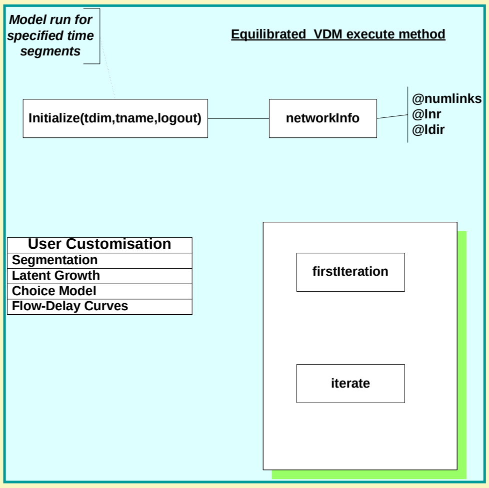


- Warning: the following slides contain design diagrams and Ruby script extracts
  - actual scripting, some extraneous elements removed
- (Experience shows scripting readily assimilated by model developers, minimal knowledge required of users)

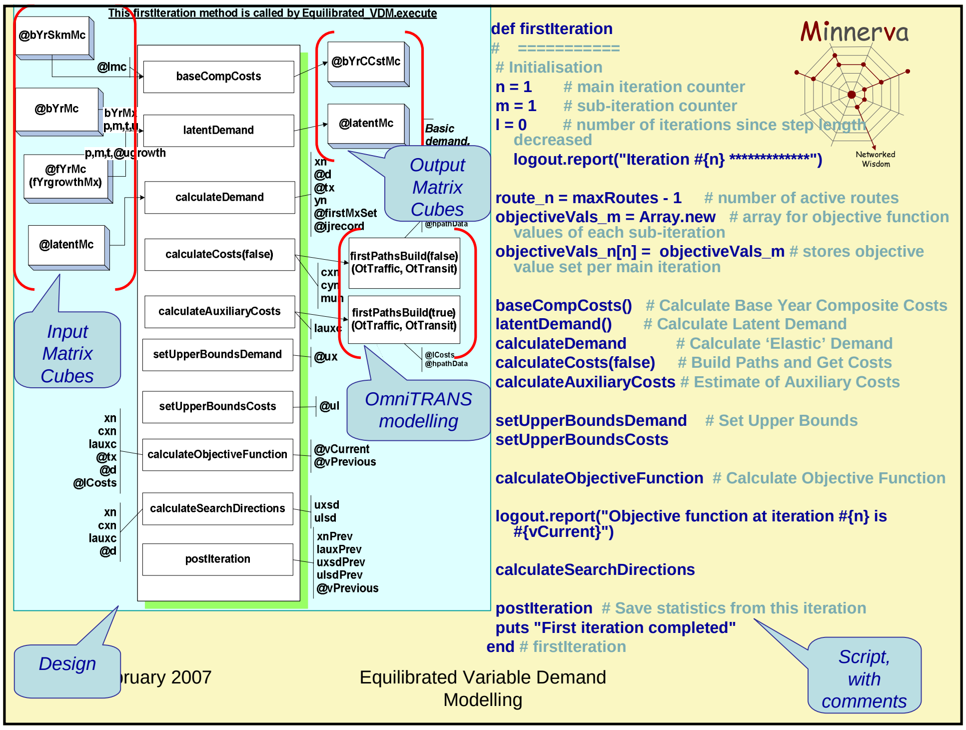
9th February 2007 Equilibrated Variable Demand Modelling

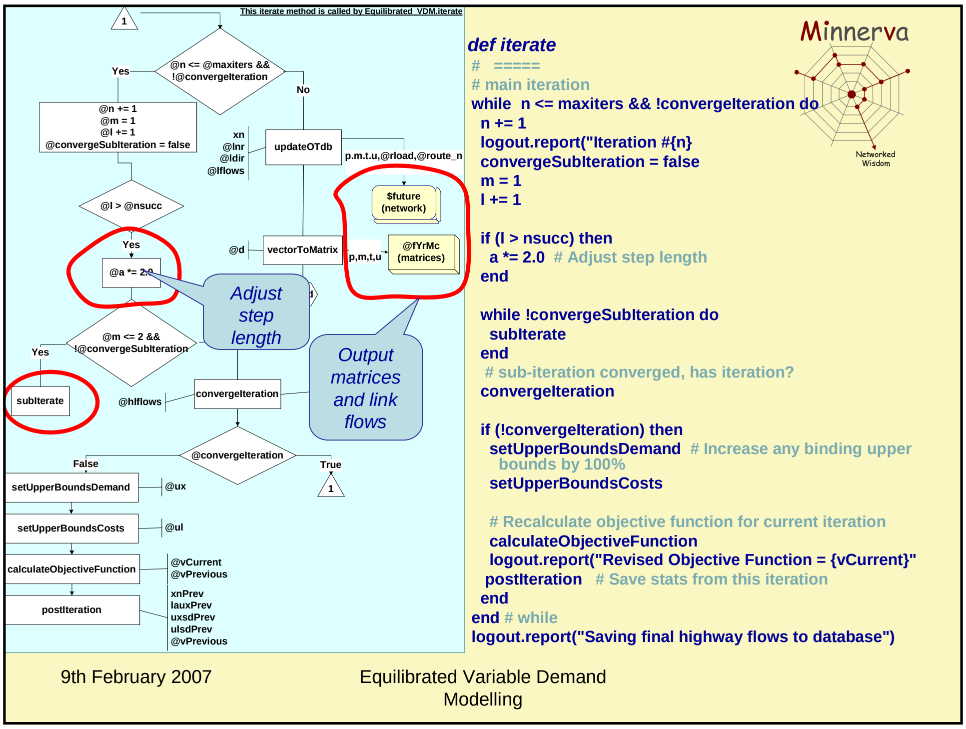
Structure of Implementation

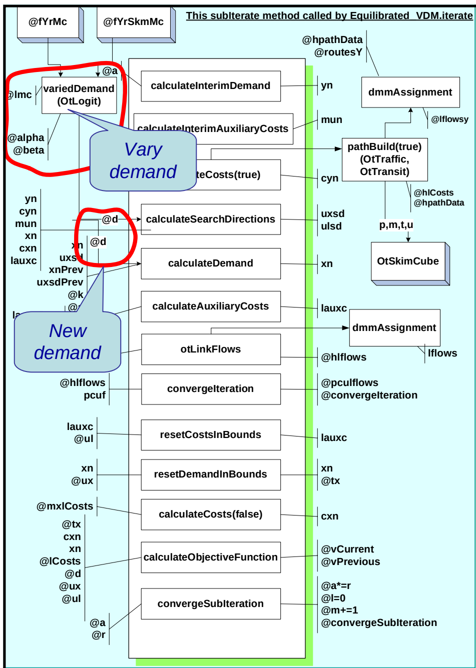
- Users supply:
  - Networks
  - Segmentation
  - Trip ends / trip end modelling
  - Choice model structure
  - Cost functions
- firstIteration
  - 'getting started'
- iterate
  - finding equilibrium solution



9th February 2007 Equilibrated Variable Demand Modelling


This firstIteration method is called by Equilibrated_VDM.execute
@bYrSkmMc
@lmc
@bYrMc
@fYrMc (fYrgrowthMx)
@latentMc
p,m,t,@ugrowth
baseCompCosts
latentDemand
calculateDemand
calculateCosts(false)
calculateAuxiliaryCosts
setUpperBoundsDemand
setUpperBoundsCosts
calculateObjectiveFunction
calculateSearchDirections
postIteration
@bYrCCstMc
@latentMc
Basic demand.
Output Matrix Cubes
firstPathsBuild(false) (OtTraffic, OtTransit)
firstPathsBuild(true) (OtTraffic, OtTransit)
Input Matrix Cubes
OmniTRANS modelling
@ux
@ul
@vCurrent
@vPrevious
uxsd
ulsd
xnPrev
lauxPrev
uxsdPrev
ulsdPrev
@vPrevious
Design
Minnerva
Networked Wisdom
def firstIteration
# ===========
# Initialisation
n = 1 # main iteration counter
m = 1 # sub-iteration counter
l = 0 # number of iterations since step length decreased
logout.report("Iteration #{n} *************")
route_n = maxRoutes - 1 # number of active routes
objectiveVals_m = Array.new # array for objective function values of each sub-iteration
objectiveVals_n[n] = objectiveVals_m # stores objective value set per main iteration
baseCompCosts() # Calculate Base Year Composite Costs
latentDemand() # Calculate Latent Demand
calculateDemand # Calculate 'Elastic' Demand
calculateCosts(false) # Build Paths and Get Costs
calculateAuxiliaryCosts # Estimate of Auxiliary Costs
setUpperBoundsDemand # Set Upper Bounds
setUpperBoundsCosts
calculateObjectiveFunction # Calculate Objective Function
logout.report("Objective function at iteration #{n} is #{vCurrent}")
calculateSearchDirections
postIteration # Save statistics from this iteration
puts "First iteration completed"
end # firstIteration
Script, with comments
Equilibrated Variable Demand Modelling


This iterate method is called by Equilibrated_VDM.iterate
@n <= @maxiters && !@convergeIteration
Yes
No
@n += 1
@m = 1
@l += 1
@convergeSubIteration = false
updateOTdb
p.m.t.u,@rload,@route_n
$future (network)
@fYrMc (matrices)
vectorToMatrix
@l > @nsucc
@a *= 2.0
@m <= 2 && !@convergeSubIteration
subIterate
convergeIteration
@convergeIteration
False
True
setUpperBoundsDemand
setUpperBoundsCosts
calculateObjectiveFunction
postIteration
Adjust step length
Output matrices and link flows
Minnerva
Networked Wisdom
def iterate
# =====
# main iteration
while n <= maxiters && !convergeIteration do
n += 1
logout.report("Iteration #{n}
convergeSubIteration = false
m = 1
l += 1
if (l > nsucc) then
a *= 2.0 # Adjust step length
end
while !convergeSubIteration do
subIterate
end
sub-iteration converged, has iteration?
convergeIteration
if (!convergeIteration) then
setUpperBoundsDemand # Increase any binding upper bounds by 100%
setUpperBoundsCosts
Recalculate objective function for current iteration
calculateObjectiveFunction
logout.report("Revised Objective Function = {vCurrent}"
postIteration # Save stats from this iteration
end
end # while
logout.report("Saving final highway flows to database")
9th February 2007
Equilibrated Variable Demand Modelling



subIterate and variedDemand

Minnerva
Networked Wisdom

```
# Generate composite (log sum) costs across all passenger modes
choice_model  =  OtChoice.new
      choice_model.form       = LOGIT
      choice_model.parameters = [@lmc]
      fYrCostMx = choice_model.logSum([car_skim,pt_skim])
probability_array = choice_model.execute([car_skim,pt_skim])

     p_car = probability_array[0]  # probabilities by mode
     p_pt   = probability_array[1]

     mx_allm = latentMc.get([pdim,$all_mode,time,udim])

$mode_list.each_index{ |mi| # New demand by mode
   mcvtype = $mode_list[mi][3]   # mode cost variability type
   demandMx = mx_allm.multiply(p_car)
   demandMx = mx_allm.multiply(p_pt) if mtype == $ptName

   # Store revised forecast demand
   fYrMc.set([pdim,mdim,time,udim],demandMx)
```

Choice modelling

9th February 2007 Equilibrated Variable Demand Modelling

Some Example Choice Models

Minnerva
Networked Wisdom

```
# Doubly-constrained destination choice model
mnl = OtChoice.new
   mnl.operation     = BY_ROW             # Attributes cal... w
   mnl.combinations = [111,211]        # Production ... sions
   mnl.parameters    = [-0.05]           # Scale p... me...
   mnl.costmatrix    = [p,m,t,u,r,i]   # Cos... of each destination for each origin
   mnl.matrix          = [p,m,t,u]        # optional: output trip matrix
   mnl.costs            = [p,m,t,u,r,i]   # optional: output composite costs
   mnl.balance       = PRODUCTION   # Doubly-constrain and scale ATTR to match
                                                    PROD total
   mnl.epsilon      = 0.001            # Convergence target
prob = mnl.execute([cost])
```

Balancing to ensure doubly constrained choices

```
sk = OtSkimCube.new ;  mc = OtMatrixCube.open
totalMx = mc[p,m,t,u]

# MNL departure time choice by OD
sklist = Array.new
1.upto(3) {|time| sklist.push(sk[p,m,time,u,r,i]) }
ch = OtChoice.new
  ch.form = LOGIT
  ch.parameters = [-0.45]
prob = ch.execute(sklist)

0.upto(2) {|time| mc[p,m,time,u] = totalM...
```

Production of set of time sensitive costs

```
# HL/MNL mode choice by OD
skC = sk[p,m1,t,u,r,i]
skB = sk[p,m2,t,u,r,i]
skT = sk[p,m3,t,u,r,i]
ch = OtChoice.new
 ch.form = LOGIT
 ch.parameters = [-0.12,-0.25]
prob = ch.execute([skC,[skB,skT]])

mc[p,m1,t,u] = totalMx * prob[0]
mc[p,m2,t,u] = totalMx * prob[1]
mc[p,m3,t,u] = totalMx * prob[2]
```

Hierarchy of choices modelling given by structure

9th February 2007 Equilibrated Variable Demand Modelling

variedDemand

```
# Generate composite costs across all passenger modes
  choice_model           = OtChoice.new
   choice_model.form       = LOGIT
   choice_model.parameters = [@lmc]
   fYrCostMx = choice_model.logSum([car_skim,pt_skim])     # composite costs
  probability_array = choice_model.execute([car_skim,pt_skim])
   p_car = probability_array[0]                              # retain probabilities by mode
   p_pt  = probability_array[1]
   mx_allm = latentMc.get([pdim,$all_mode,time,udim])

   # Calculate revised demand matrix reflecting current costs
   bYrCostMx = bYrCCstMc.get([pdim,$all_mode,time,udim])  # get composite costs
   tmpMx = fYrCostMx.divide(bYrCostMx);  tmpMx.power!(@alpha) # Power elasticity
   mx_allm.multiply!(tmpMx)              # Trips reflecting cost ratios elasticity
   tmpMx = bYrCostMx.subtract(fYrCostMx); tmpMx.multiply!(@beta);  tmpMx.exp!
   mx_allm.multiply!(tmpMx)              # Trips also reflecting cost changes
   fYrMc.set([pdim,$all_mode,time,udim],mx_allm)

   $mode_list.each_index{ |mi|                        # Calculate revised demand by mode
     mcvtype = $mode_list[mi][3]                      # mode cost variability type
     demandMx = mx_allm.multiply(p_car)  if mtype == $hwayName
     demandMx = mx_allm.multiply(p_pt)    if mtype == $ptName
     bYrMx = bYrMc.get([pdim,mdim,time,udim])         # Base trip matrix
     grwthMx = latentMc.get([pdim,mdim,time,udim])   # Growth factored from base trip matrix

     fYrMc.set([pdim,mdim,time,udim],demandMx)       # Store new forecast demand
     sc=OtSkimCube.open
     skm = sc[pdim,mdim,time,udim,rcost,idim]   # costs skim
     @d = matrixToVector(demandMx, d, pdim,mdim,udim)  if mcvtype == 'CostVaries'
    } # mode
   } # user
  } # purpose
 return @d  # demand matrix in vector form
end # variedDemand
```

Choice modelling

Elasticity modelling

New demand

Returns consolidated (demand) information

9th February 2007 Equilibrated Variable Demand Modelling

Objective Function

```
def calculateObjectiveFunction
#   ========================

@vPrevious = @vCurrent
v = 0.0

0.upto(@route_n ){|rn|
  cxn = @routesCx[rn]
  xn = @routesX[rn]

  xn.each_index{|s|
  # Route-based component of objective function
    lam = @lCosts[s]
    c = cxn[s]
    x = xn[s]
    v += (x * x) * (positive(c - lam))**2
    v += ((@ux[s] - x)**2) * (positive(lam - c))**2
  }
} # upto @route_n

0.upto(@vlen){|s|
  # Non-route-based component of objective function
  lam = @lCosts[s]
  x = @tx[s]
  y = @d[s]  # demand calculated in  variedDemand
  v += lam*lam * (positive(x - y))**2
  v += ((@ul[s] - lam)**2) * (positive(y - x))**2
}

@vCurrent = v
return @vCurrent
end # calculateObjectiveFunction
```

Network flows

- Comprehensive:
  - Matrix Demand (ij)
  - Network Flows (ijr)
  - Segments (pm[t]u)

Matrix demand

9th February 2007 Equilibrated Variable Demand Modelling

Running Equilibration Model

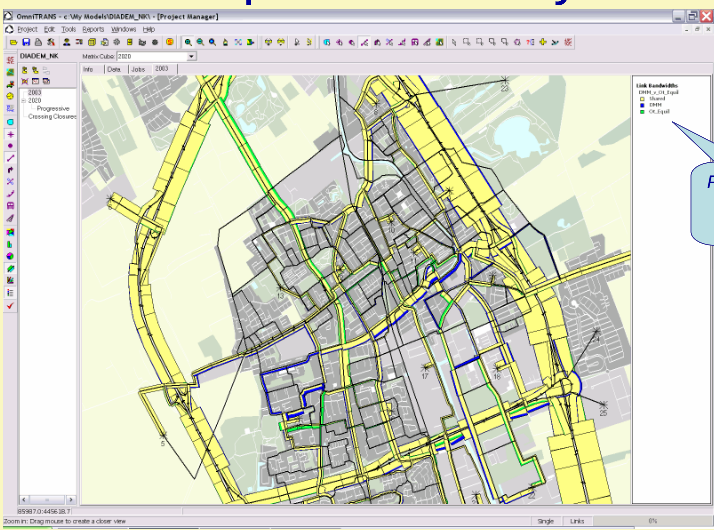
- Functionality provided in a class (library) – 'Equilibrium'
- User over-rides defaults, as required:
  - e.g. for precision of solution
- Information on coefficient values etc provided in a parameters file
- User runs job run by clicking on the script in the 'Jobs' tab

```
logout.report("
 Variable Demand Equilibrium Modelling
 =============================")

$time_list.each_index{ |ti|
  time  = $time_list[ti][0]
  tname = $time_list[ti][1]

  dmm = Equilibrium.new(time, tname, logout)
    dmm.maxRoutes = 3
    dmm.cnvgPercentLnkTolerance = 2
    dmm.dsgap_target = 0.2
  dmm.execute

} # time_list

 logout.close
```

User (precision) controls

9th February 2007 Equilibrated Variable Demand Modelling

Some Comparative Analyses



Results using different methods

9th February 2007 Equilibrated Variable Demand Modelling

Summary

- VDM needed for improved decision making
  - More reliable model results
  - Now part of formal DfT guidance
- OmniTRANS implementation
  - Flexible and straightforward modelling framework
  - Concise scripting
    - Open to inspection and adaptation
  - Makes VDM approachable for different types of studies
- Full-scale application due
  - Proving trials on-going
  - Scope for further optimisation of operation and for different solution methods