2019 User Group Meeting
SATURN 102: Part 2 - Skimming

November 2019

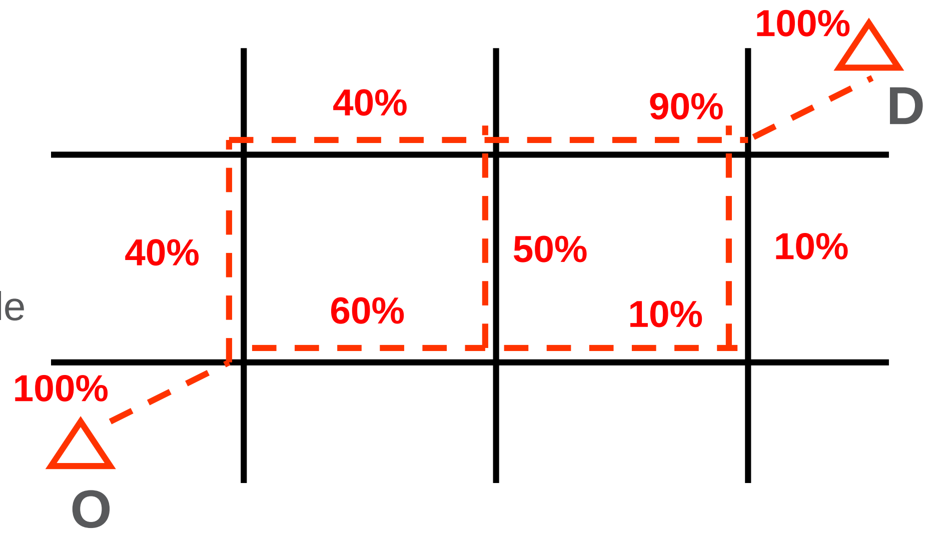
UNIVERSITY OF LEEDS

FVVB Ltd

ATKINS
Member of the SNC-Lavalin Group

SATURN 102 Part 2 - Skimming

- Following on from Part 1, skimming uses the same paths as Matrix Estimation
  - The difference is the extraction of **cost information** rather than **demand**
    - Either in terms of generalised cost or its components (ie time, distance, tolls, penalties)
  - Same requirement to recreate paths from the stored link data in the UFC file
    - Uses **SATLOOK** rather than **SATPIJA**



Need a little recap before discussing options

- This Year's SATURN 102
  - Previous Matrix Estimation discussed paths used to update matrix
- Last Year's SATURN 101
  - Part 1 – SATURN Capacities described the SATASS – SATSIM looping Process
  - Part 3 – Convergence including SAVEIT approximation

Assigned Paths

- From the SAT102 previous session

Background Essentials (ii)

Recap

Assignment:

- Single All-or-Nothing (AON) - allocates all the OD-demand to a single route (or 'path')

Equilibrium Assignment

- Series of AON assignments with paths costs varying through capacity constraint, leading to:

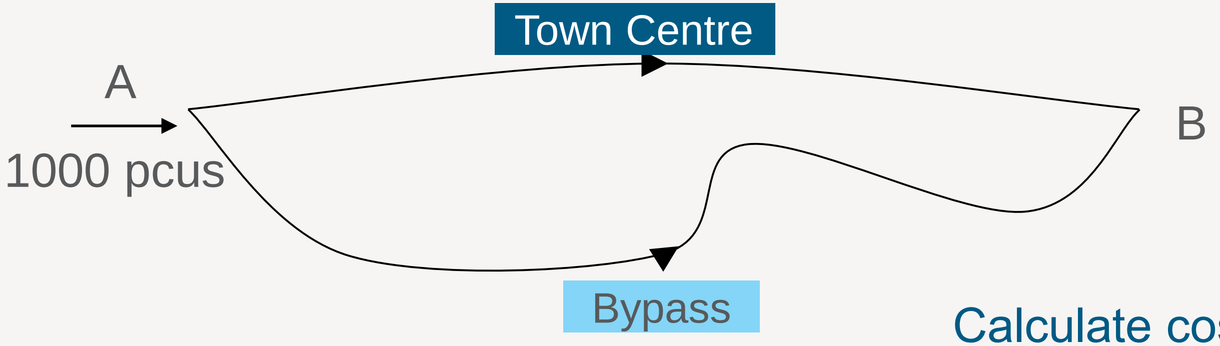
Wardrop Equilibrium

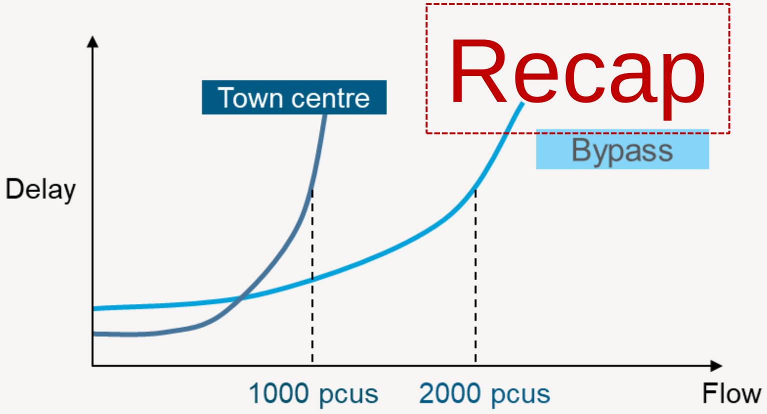
- "Traffic arranges itself on networks such that the cost of travel on all routes used between OD pair is equal to the minimum cost of travel and all unused routes have equal or greater cost" (TAG Unit M3.1)

In SATURN, this mathematical process is undertaken by:

- 'minimising' an objective function
- using the Frank-Wolfe algorithm
- to determine the optimum combination (lambda) of the available AoN assignments.

Assignment for a Buffer Network (ii) - Combining AoN solutions for Equilibrium





Path-building: Successive All-or-Nothings

… allocate 1000 pcus to either Town Centre or Bypass

Path Build	Town Centre	Bypass
1	1000	0
2	0	1000
3	0	1000
4	1000	0

Calculate costs based on flows of …

Iteration	Town Centre	Bypass
1	1000	0
Combine (e.g. 0:100)	(1000)	(0)
2	0	1000
Combine (e.g. 50:50)	(500)	(500)
3	0	1000
Combine (e.g. 66:33)	(333)	(667)
4	1000	0

Accumulating the final set of paths

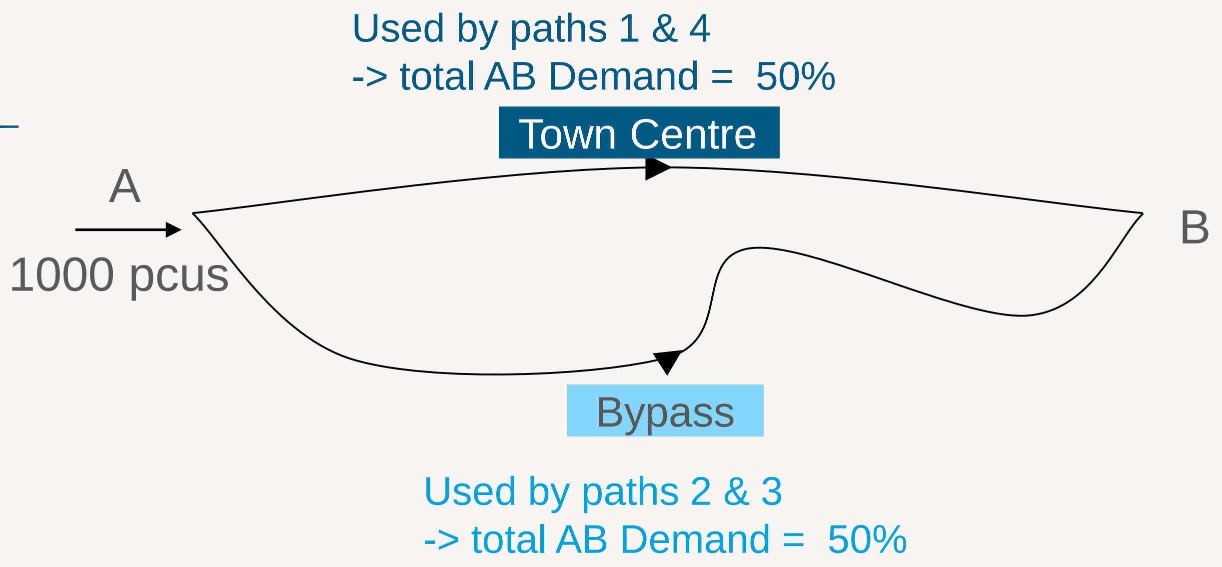
Recap

Having calculated the costs based on flows of …

Iteration	Town Centre	Bypass
1	1000	0
Combine (e.g. 0:100)	(1000)	(0)
2	0	1000
Combine (e.g. 50:50)	(500)	(500)
3	0	1000
Combine (e.g. 66:33)	(333)	(666)
4	1000	0
Combine (e.g. 75:25)	(500)	(500)

Now visualise the forest between A, B

- As Method of Successive Averages used, equal weight attached to each iteration
- Link costs based on final combined flows



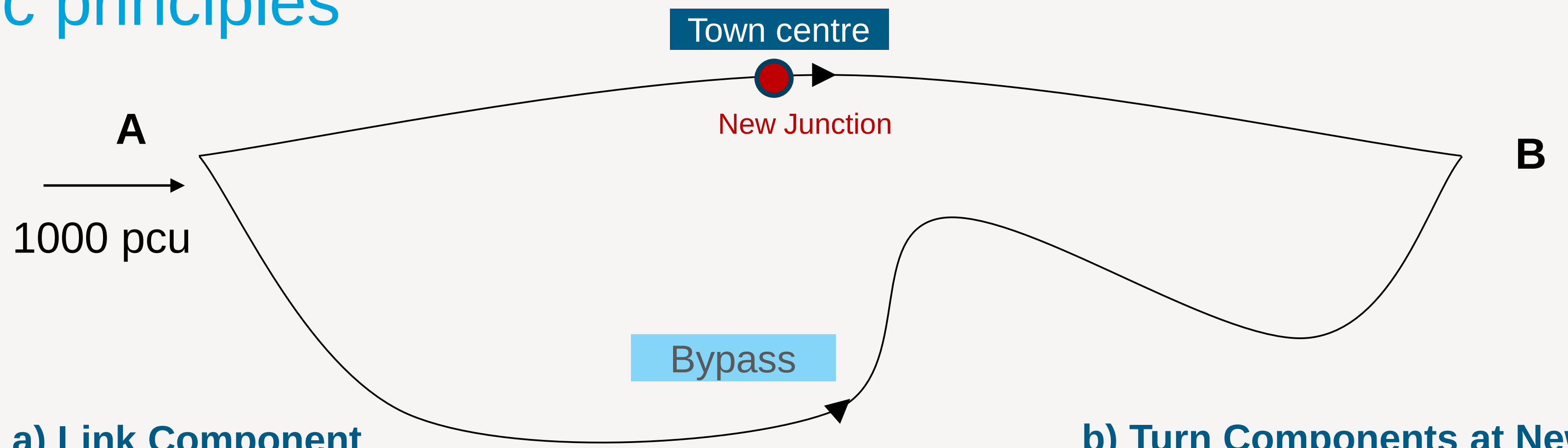
Assignment & Simulation

- From the SATURN 101 last year !

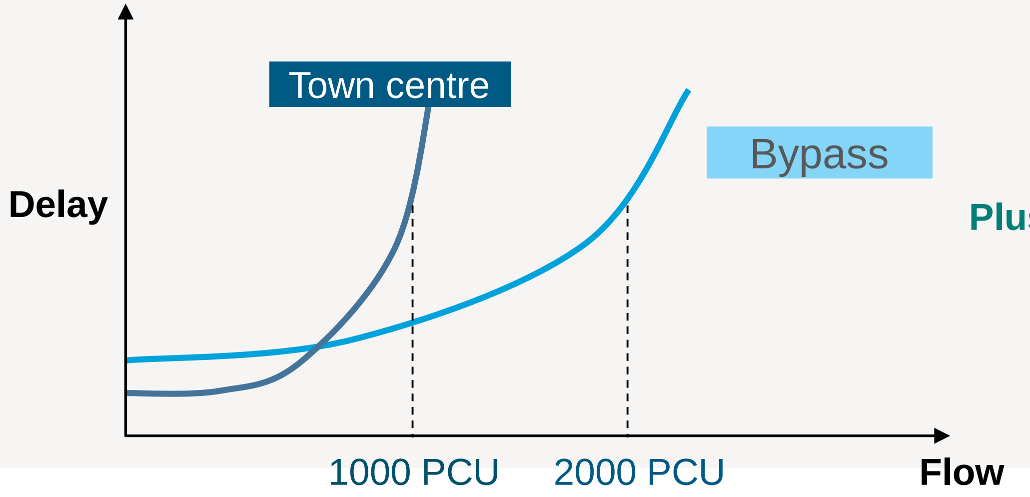
ATKINS
Member of the SNC-Lavalin Group

Assignment for a Simulation Network (i) - Basic principles

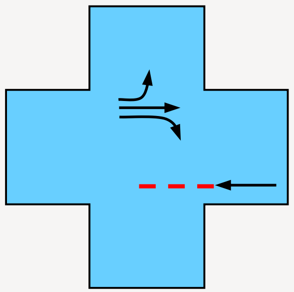
Recap



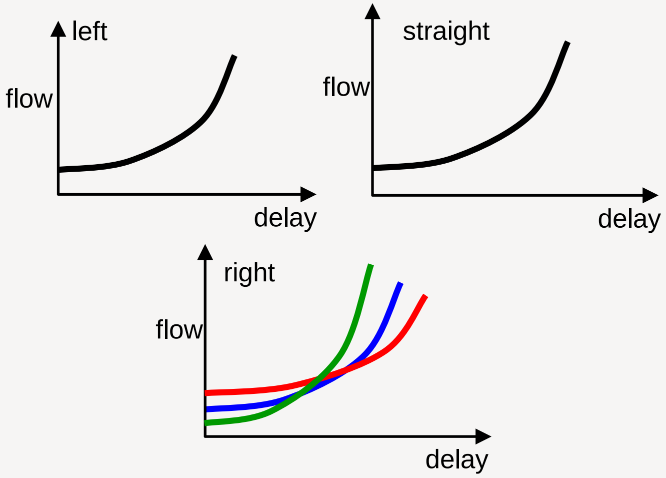
a) Link Component



Plus



b) Turn Components at New Junction



SATURN Assignment & Simulation

Recap

Assignment sub-model (SATASS)

› uses the ‘assignment’ network =

Buffer Network + *Exploded Simulation Network*

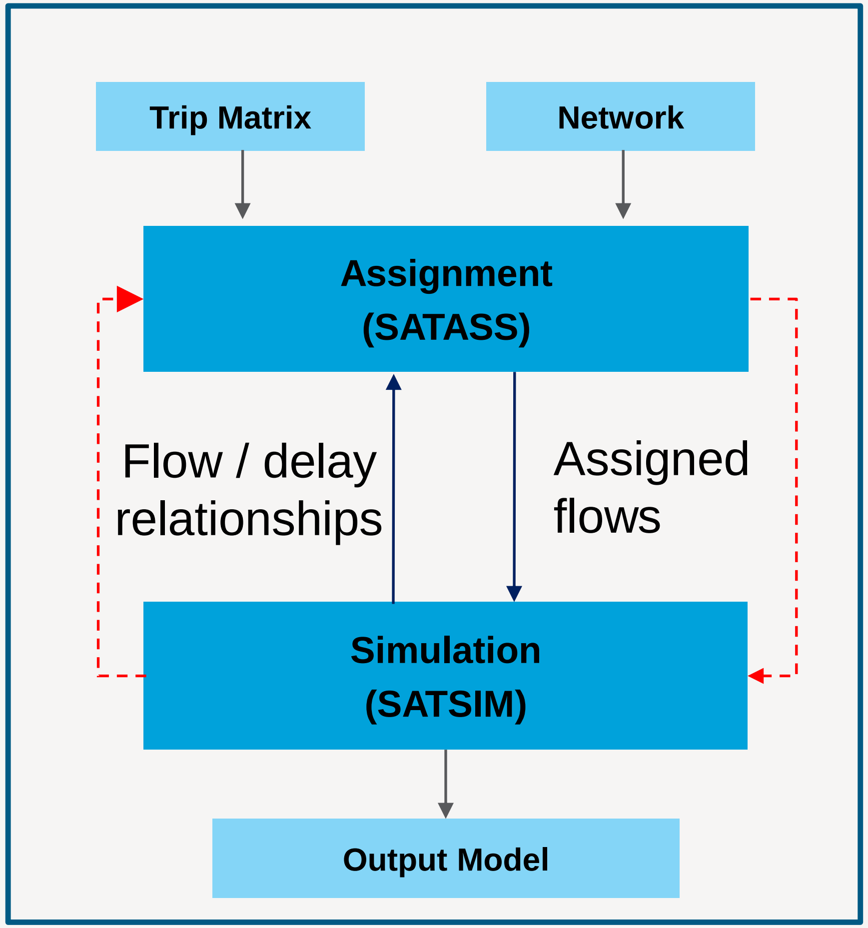
In terms of the assignment, there is no distinction between the two – each has its own flow-delay curve

But … their flow-delay curves have been generated by two different processes:

› Buffer = explicitly defined by the users
› Simulation = generated by the **SATURN Simulation (SATSIM)**

Iterative process until convergence achieved

› Both within SATASS & SATSIM
› AND also between successive ASS-SIM loops



Recap

SATURN Assignment 101 - Assignment Process

Flows →

Loop	Assignment	Simulation	Cumulative (AON) Paths
1	20 Path Builds	20 Simulation Iterations	0 + 20 = 20
2	20 Path Builds	20 Simulation Iterations	20 + 20 = 40
3	20 Path Builds	20 Simulation Iterations	40 + 20 = 60
…			
n	20 Path Builds	20 Simulation Iterations	= n * 20

← Flow / Delay Curves

Hence: converged assignment solution consists of up to # of Loops * 20: a huge number?

SAVEIT Approximations

Recap

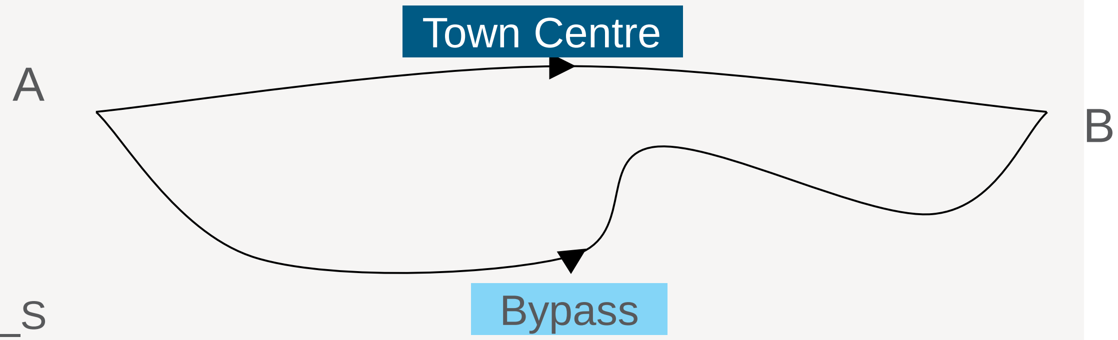
Cost data stored in the UFC file for secondary analysis

Recreates assignment using either :

- the original full set of paths used OR a SAVEIT approximation

By default, UFC109=T & NITA_C=256 so

- full set saved unless cumulative path builds > 256
- otherwise SAVEIT used - maximum no. of path builds set by NITA_S



Value of NITA_S is very important

- If too small (e.g. 25!) then too few paths used in SAVEIT approximation
- Likely that very poor Wardrop solution (Approximation %GAP >> Final %GAP)
- Use v11.4 default: NITA_S=256 is sensible

Support feedback:

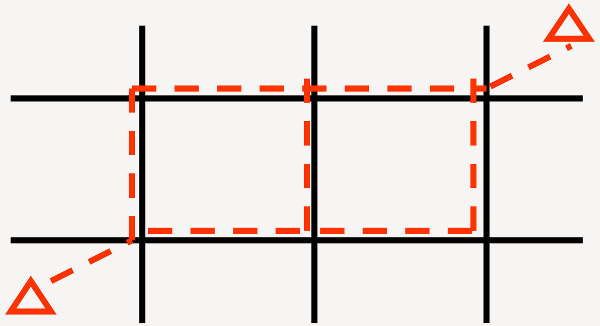
- Models with very large values of NITA_C or NITA_S (eg > 600)
- Not required – check what's required!
  - *very large UFC files, significant extra CPU for SAVEIT and long runtimes for secondary analysis*

The Skimming Process

- Extracting OD costs

ATKINS
Member of the SNC-Lavalin Group

Skimming Options Available



Core Processes

Process	Parameter	Description	Type	Multi-Core	Comments
Batch	SATCOST	Build AON trees based on a cost and report the minimum costs between zones.	Final Path + 1	No	Fast as single path skimmed
Batch	SKIMDIST / TIME / TOLLS / PEN	Skim components of generalised costs (ie time, distance, toll, penalty)	Demand weighted average	**Yes**	Slower as paths need to be recreated QUICK - skims final path only QUICK *n* – skims Top *N* paths
Batch	SKIM_ALL	As above but skimming all four at the same time	Demand weighted average	**Yes**	
Batch	SKIMDA	Skim specific DA code	Demand weighted average	**Yes**	
SATLOOK	Manually	Replicate above plus other options	Option specific	No	

Command Line Overrides

Parameter	Description	Defaults	Comments
USESPI	Skims using the SPIDER sub-network rather than full network	T	Faster as more efficient sub-network structure
USEUFO	Skims using the UFO file rather than the UFC	As defined in Network File	
NOT_USE*val*	Inverts the selection eg: NOT_USEUFO forces UFC etc	N/A	Can become complicated – see 15.27.7.5

A Case Study – HAM P4 Tests using QUICK*n*

Skimming Issues

Equilibrium solution produces minimum generalised costs

- Component skims (ie time, distance, tolls, penalties) are not unique!

Minimum (Demand) Cost as good as / better than average

Skimming a single path inaccurate if link-specific tolls / area charges

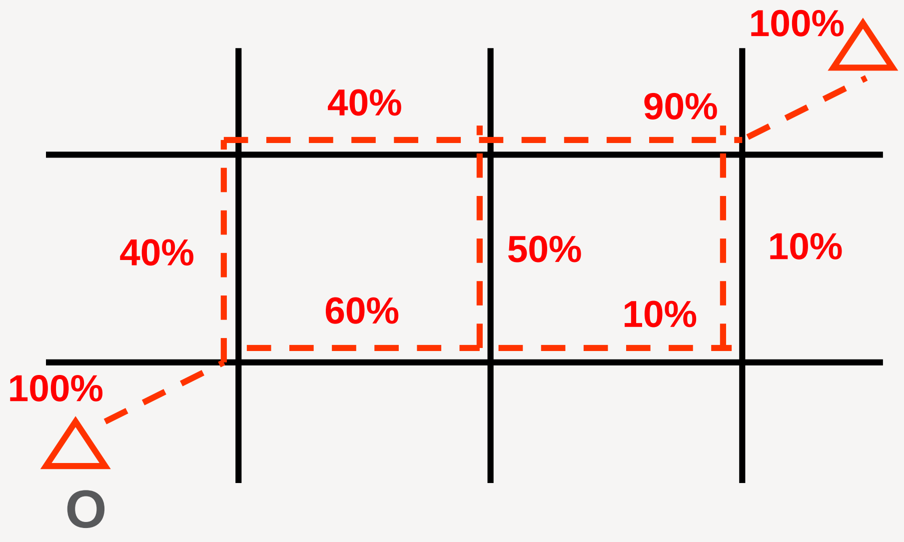
Links to Demand Models

- Common that skimmed components are used to define alternative demand costs
  - Dangerous & process inherently non-convergent

Skimming all paths is CPU expensive

- Investigations undertaken to find faster methods
- Use SATGPU?

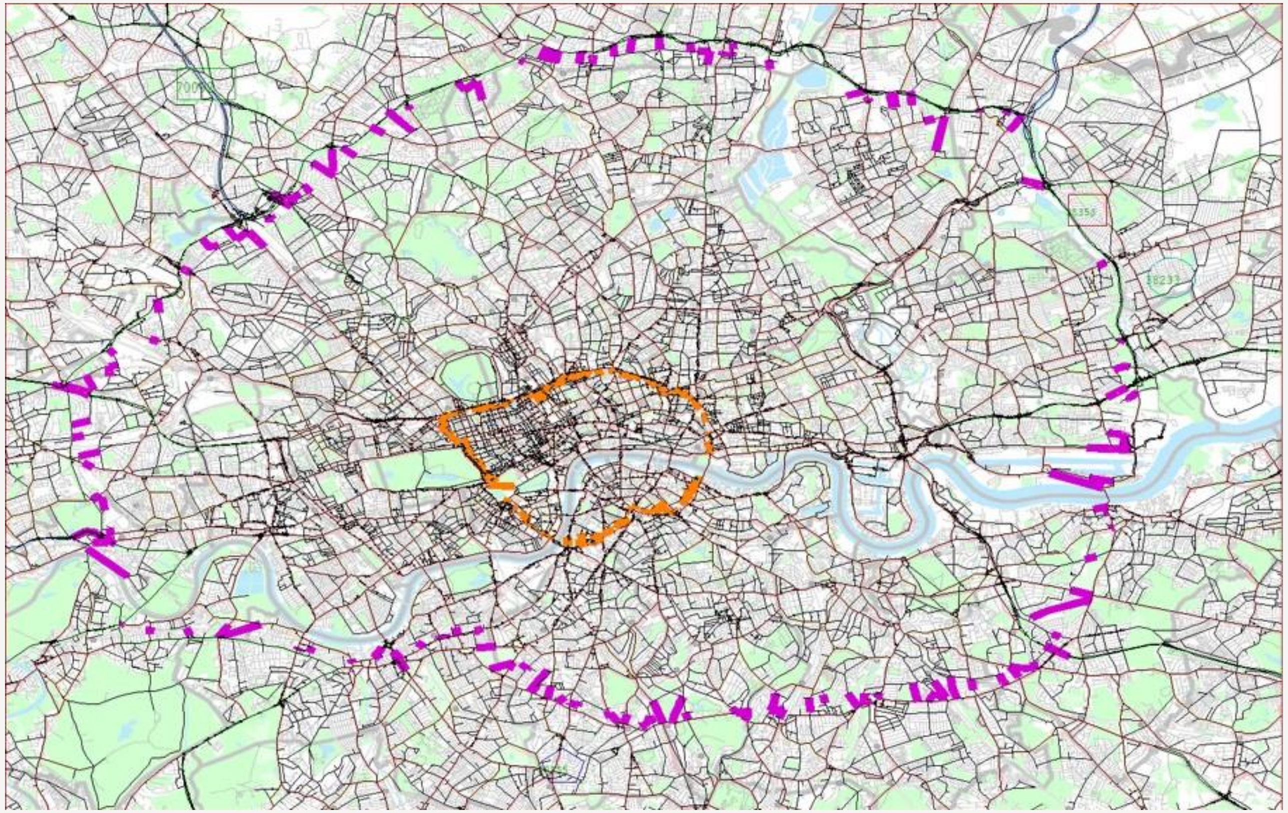
New QUICK *N* option developed for SATURN 11.5 TfL Area Charging



A Case Study – QUICK N ?

TfL HAM P4 forms part of the new MoTION modelling system

- Investigations into model runtimes including:
  - CASSINI
  - Faster assignments
    - *Dijkstra, SATALL parameters etc*
  - SATGPU
  - Reduce skimming times
    - *New QUICK n option*
- Undertake skim comparison
  - FULL versus QUICK versus QUICK n
  - Using HAM P4 beta
    - *Differences in GC Skims*
    - *Westminster -> Camden & City*
    - *UC2 Time*



Distribution of Assigned Top Paths versus Assigned Demand

%Demand assigned to each path is not equal

- Determined by Frank-Wolfe algorithm (lambda values)
- Observed tendency for later paths to have more demand assigned

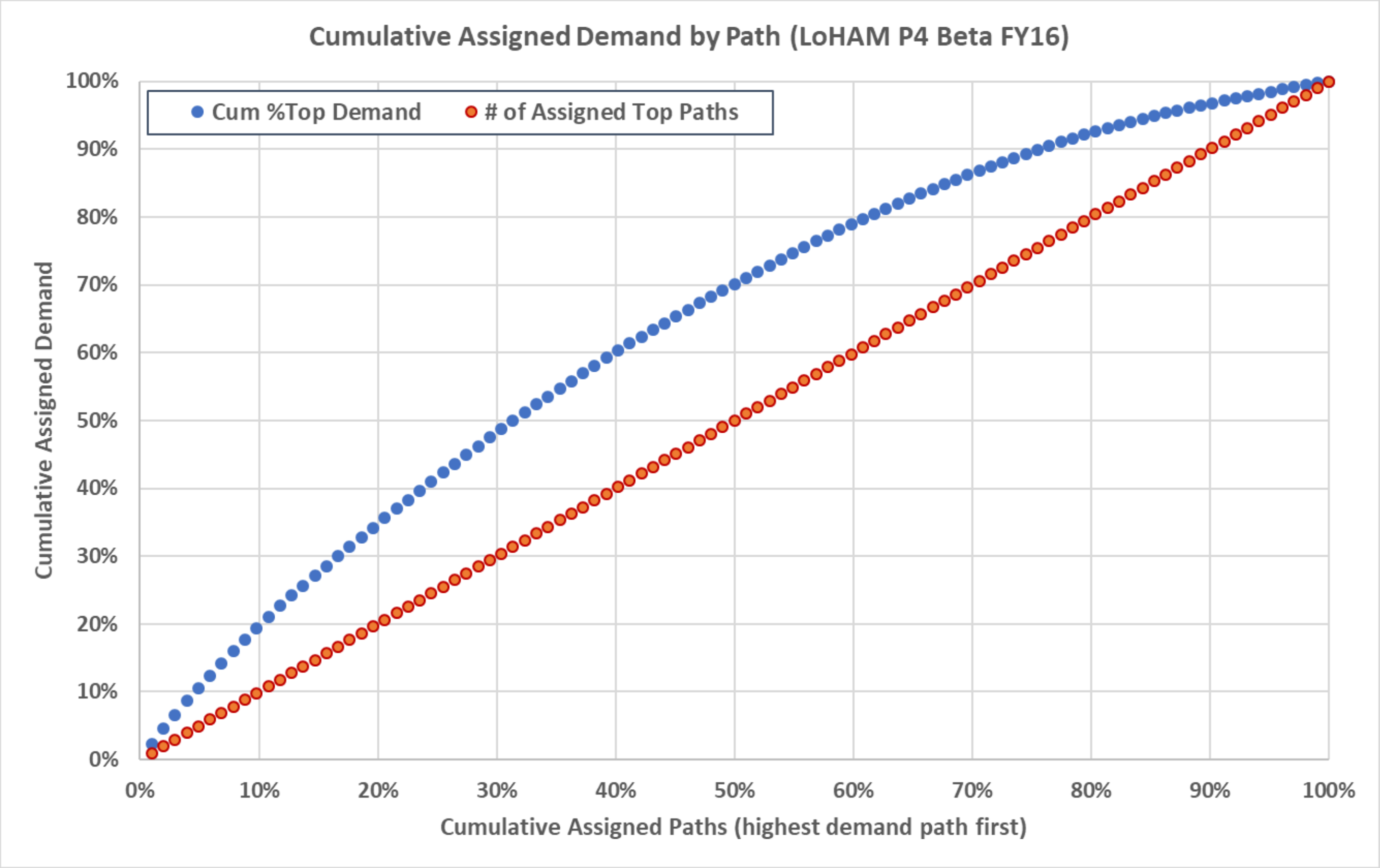
CPU time savings available if only skimming the most heavily used (‘Top’) paths

- Linear CPU time for path skimming

New QUICKn option

- Skims Top ‘N’ paths: user defined
  - Latest LPT reports on distribution

How many paths to skim?



ATKINS
Member of the SNC-Lavalin Group

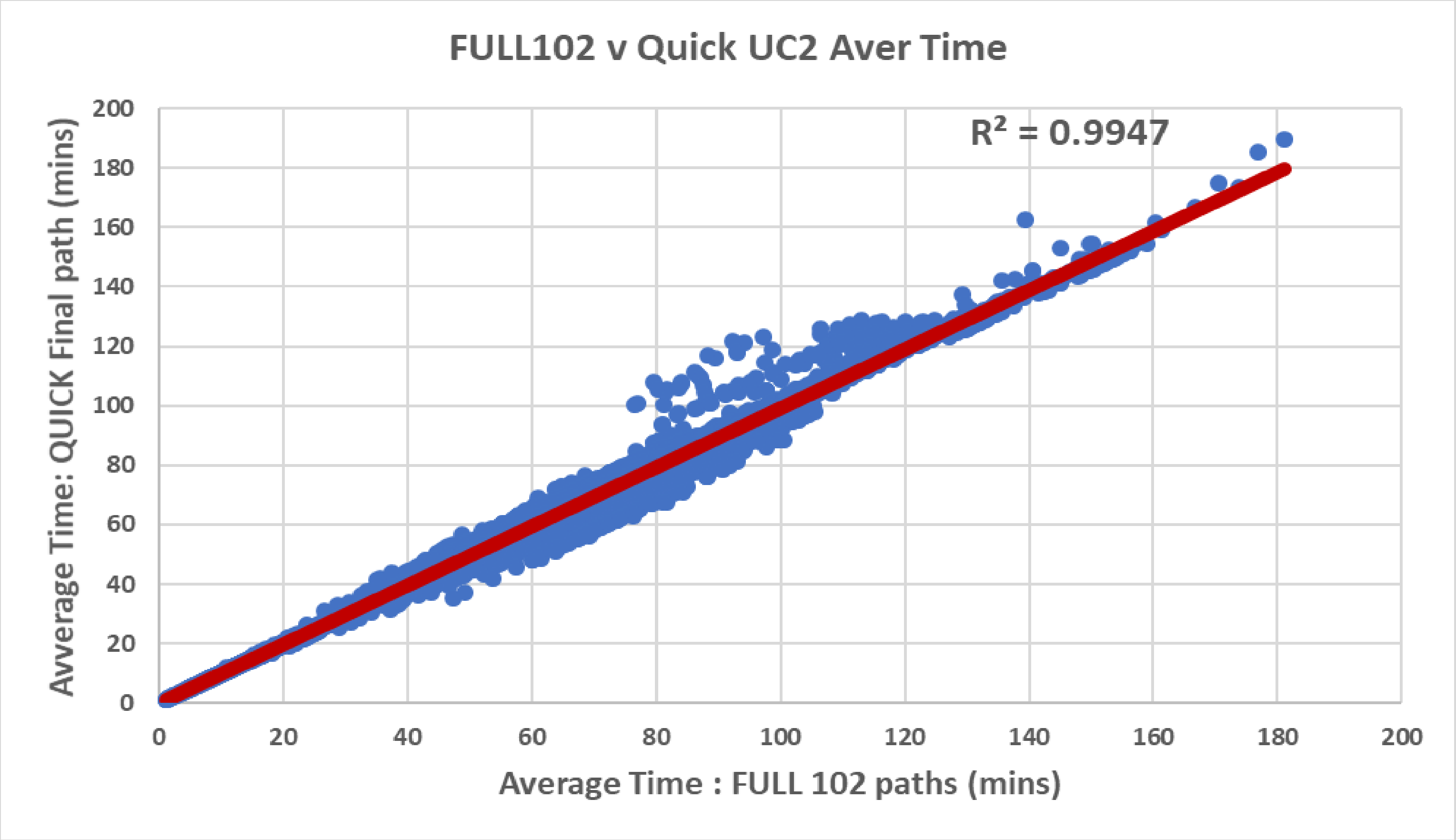
SATURN

Final 30/11/19

Comparisons: UC2 Average Time QUICK v FULL (Absolute)

Comparing:

- Full = All 102 paths
  - 100% demand
- Quick = Final path
  - Via ‘QUICK’ option
- Highlights differences
  - Difficult to draw conclusions
  - Replot using error distribution

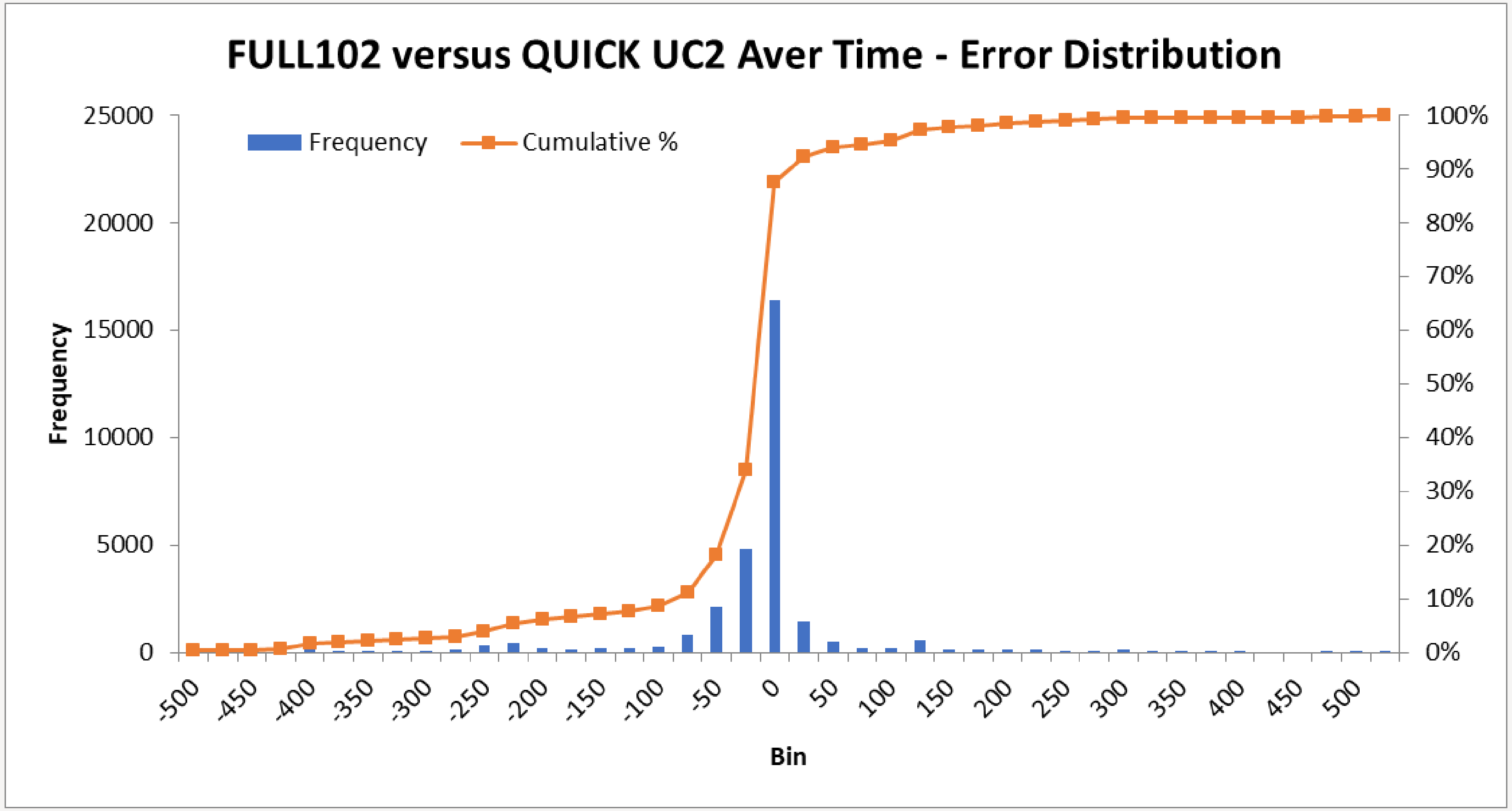


Comparisons: UC2 Time QUICK v FULL (Error Distribution)

Comparing:

› Error Distribution

Measure	Value
Assignment	
# of Paths	QUICK
%Demand	FINAL
%Skim Time	1%
Errors (mins)	
Mean	-0.5
Min	-14
Max	29
Std Dev	1.8

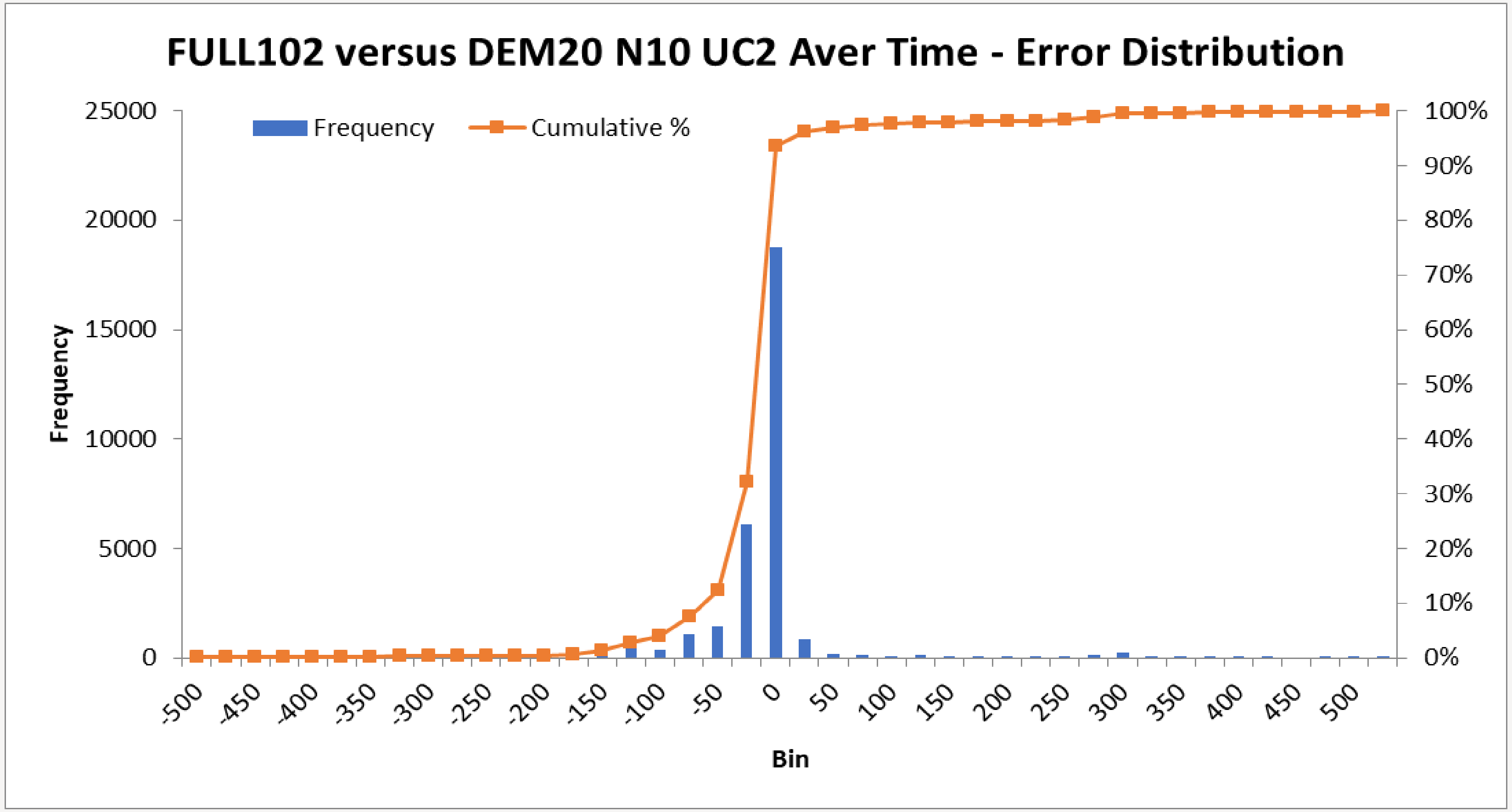


Comparisons: UC2 Time QUICK v 20% Demand (Error Dist.)

Comparing:

› Error Distribution

Measure	Value
Assignment	
# of Paths	10 / 102
%Demand	20%
%Skim Time	10%
Errors (mins)	
Mean	-0.3
Min	-14
Max	+17
Std Dev	1.2

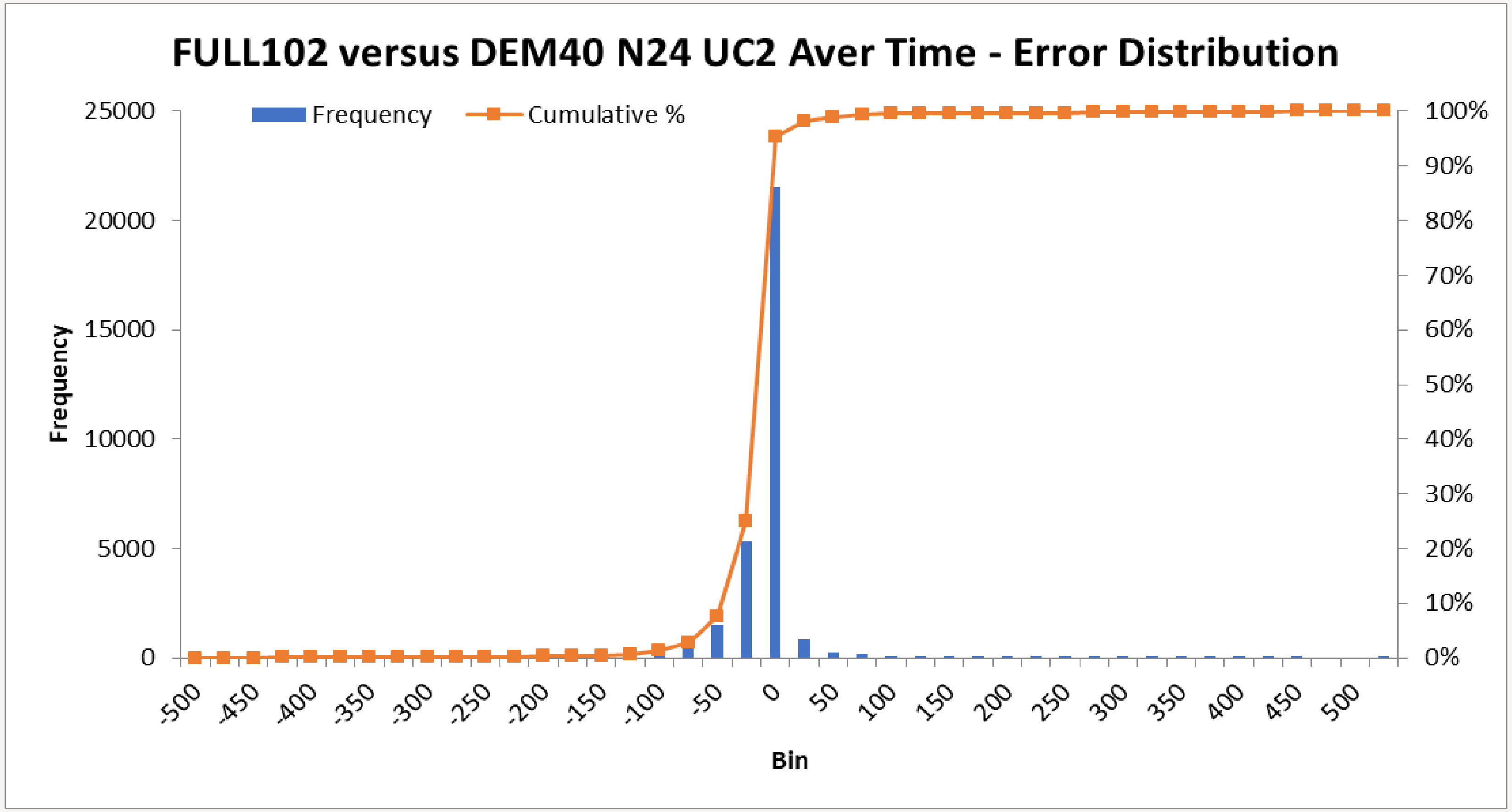


Comparisons: UC2 Time QUICK v 40% Demand (Error Dist.)

Comparing:

› Error Distribution

Measure	Value
Assignment	
# of Paths	24 / 102
%Demand	40%
%Skim Time	24%
Errors (mins)	
Mean	-0.3
Min	-8
Max	+9
Std Dev	0.6

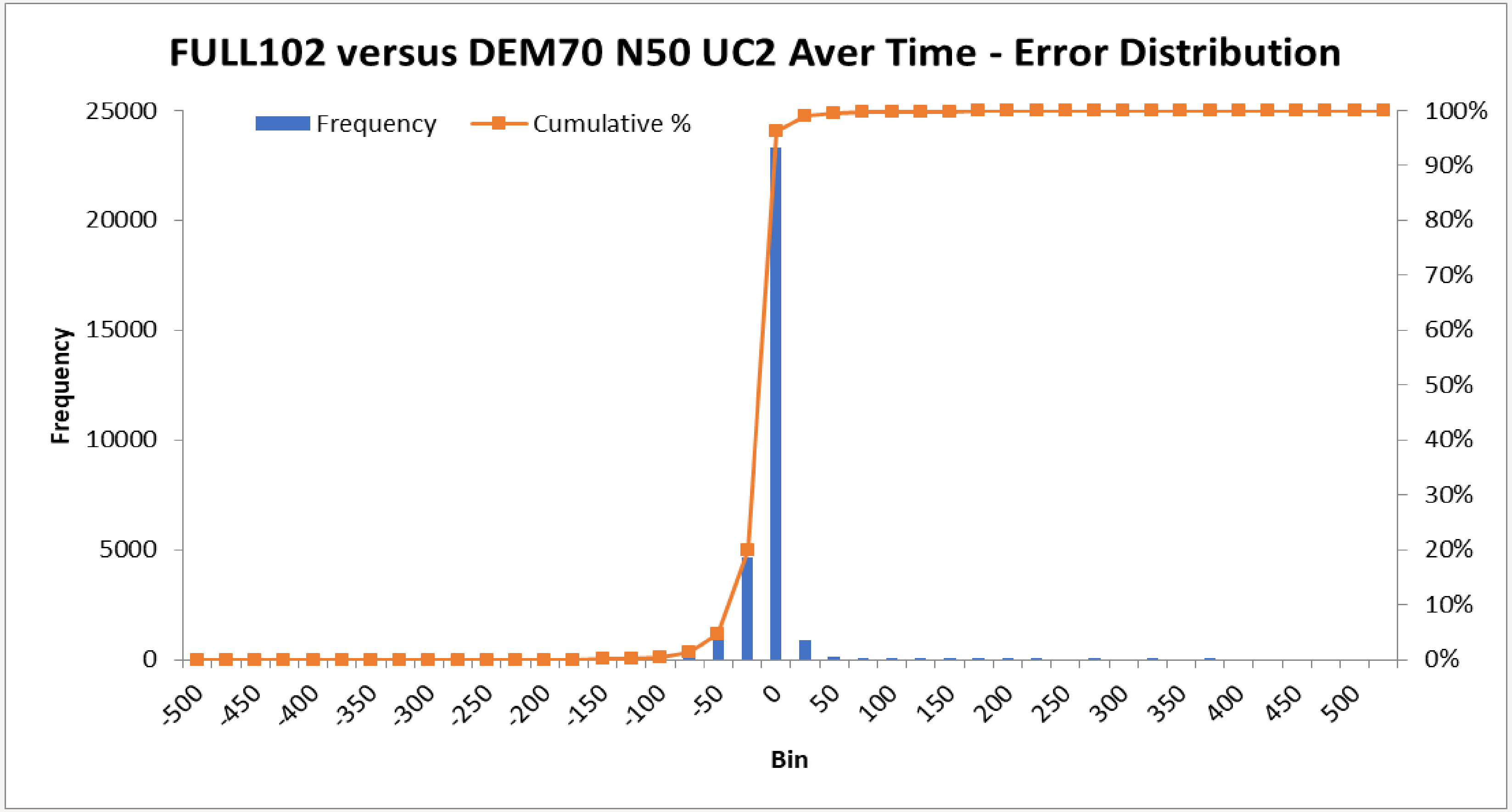


Comparisons: UC2 Time QUICK v 70% Demand (Error Dist.)

Comparing:

› Error Distribution

Measure	Value
Assignment	
# of Paths	50 / 102
%Demand	70%
%Skim Time	49%
Errors (mins)	
Mean	-0.3
Min	-3
Max	+6
Std Dev	0.3



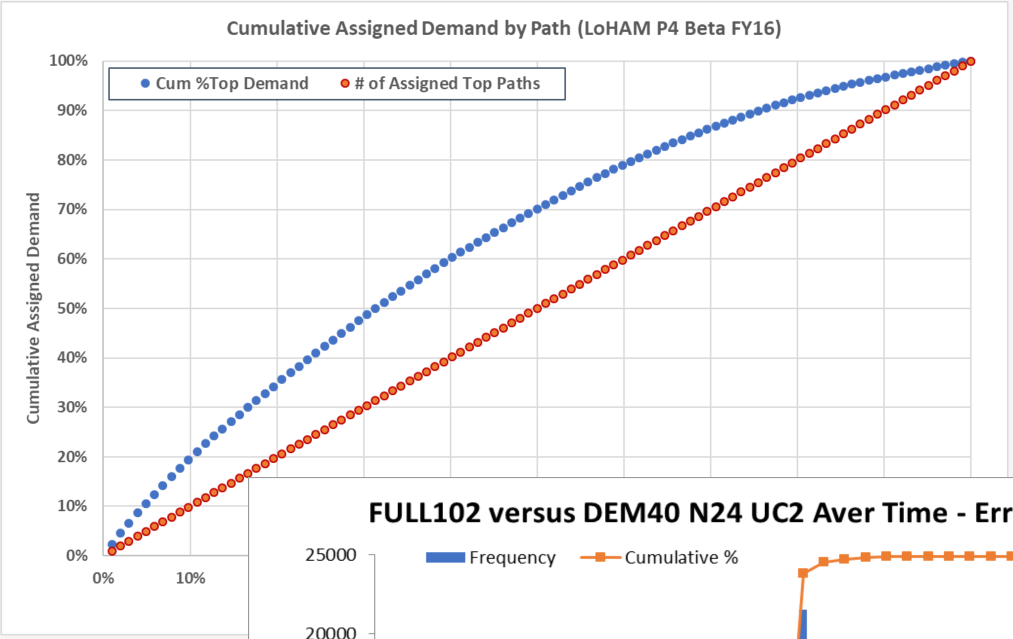
Emerging Findings

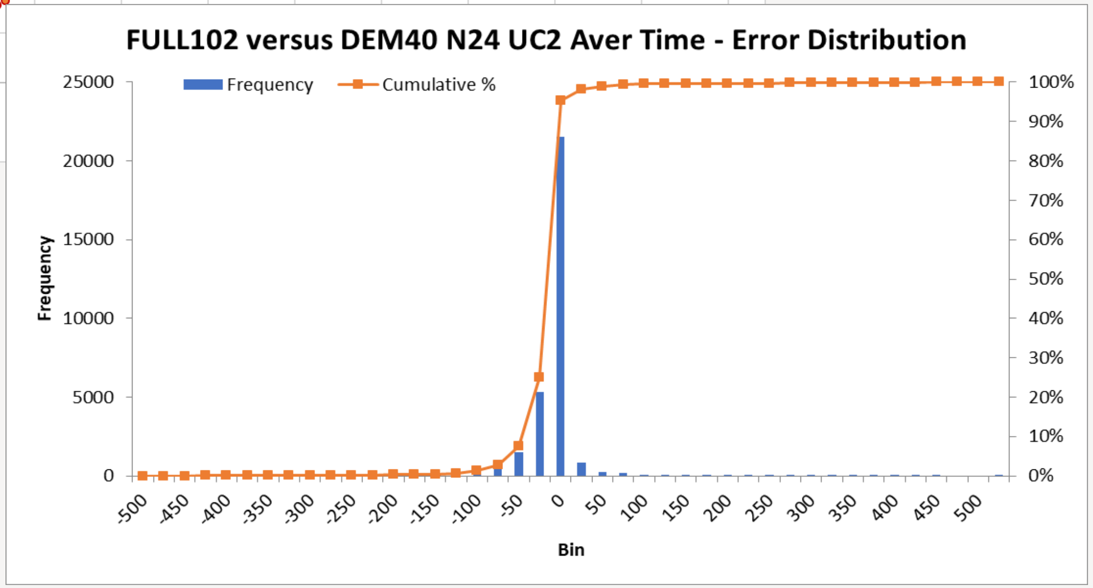
Looks promising

- Difficult to draw definitive conclusions – are the differences important?
- Further investigations required within MoTION to determine whether material impact on model convergence

Confirms that faster model runtimes are readily achievable

  - Continued role for CASSINI-based approaches
  - Complimentary to SATGPU-based techniques





Questions

Supplementary Information

Step 3b – Checking SAVEIT Performance

Reports in the LPT file

- Compares accuracy of main assignment versus SAVEIT
  - Take %Epsilon rather than %Delta

Bad Example: %Epsilon = 0.1743%

```
             WARDROP MUC USER EQUILIBRIUM ASSIGNMENT


 TREE BUILDING AND LOADING ALGORITHMS ARE BASED ON A SPIDER WEB
 AGGREGATION OF NETWORK NODES AND LINKS.


 >>>>> REASSIGNMENT STOPPED AFTER   25 ITERATIONS >>>>>
       MAXIMUM NUMBER OF ITERATIONS NITA EXCEEDED


 FINAL CONVERGENCE STATISTICS AND STOPPING VALUES

       25 GE         25 - NUMBER OF ITERATIONS (<NITA)
    11.11 LT       0.05 - % OF NEW A-O-N LOAD USED (<XFSTOP)
   0.1644               - % DELTA (ACTUAL COSTS LESS MINIMUM COSTS)
    0.019               - % CHANGE IN TOTAL TRAVEL COSTS (LAST ITER)
   0.1743 LT     0.0098 - % EPSILON: UNCERTAINTY IN THE OBJ. FUNCTION (<UNCRTS)
                          (RELATIVE TO THE OBJECTIVE FUNCTION)
    1.983 LT       0.05 - % REDUCTION IN THE UNCERTAINTY (<FISTOP)
                          (RELATIVE TO THE UNCERTAINTY)
    0.010               - % REDUCTION IN THE OBJ. FUNCTION
                          (RELATIVE TO THE OBJECTIVE FUNCTION)

         0.397465E+09 - FINAL OBJECTIVE FUNCTION VALUE
```

Good Example: %Epsilon = 0.0098%

```
             WARDROP MUC USER EQUILIBRIUM ASSIGNMENT


 TREE BUILDING AND LOADING ALGORITHMS ARE BASED ON A SPIDER WEB
 AGGREGATION OF NETWORK NODES AND LINKS.


 >>>>> SAVEIT CONVERGENCE ACHIEVED AFTER  150 ITERATIONS >>>>


 FINAL CONVERGENCE STATISTICS AND STOPPING VALUES

      150 GE        256 - NUMBER OF ITERATIONS (<NITA)
     1.03 LT       0.05 - % OF NEW A-O-N LOAD USED (<XFSTOP)
   0.0103               - % DELTA (ACTUAL COSTS LESS MINIMUM COSTS)
    0.000               - % CHANGE IN TOTAL TRAVEL COSTS (LAST ITER)
   0.0098 LT     0.0098 - % EPSILON: UNCERTAINTY IN THE OBJ. FUNCTION (<UNCRTS)
                          (RELATIVE TO THE OBJECTIVE FUNCTION)
    0.823 LT       0.05 - % REDUCTION IN THE UNCERTAINTY (<FISTOP)
                          (RELATIVE TO THE UNCERTAINTY)
    0.000               - % REDUCTION IN THE OBJ. FUNCTION
                          (RELATIVE TO THE OBJECTIVE FUNCTION)

         0.397085E+09 - FINAL OBJECTIVE FUNCTION VALUE
```

Impact on TUBA Scheme Appraisal - Illustrative Example

Two Scenarios (With & Without Scheme), 60 year appraisal

	Ref Case	Run 1	Run 2	Run 3	Run 4	Run 5
NITA_S	256	25	99	256	256	256
NISTOP	4	4	4	5	4	4
RSTOP	98.5%	98.5%	98.5%	98.5%	97.5%	94.5%
AM - %Flow	98.9%	98.9%	98.9%	98.5%	98.0%	96.7%
AM - %GAP (Main)	0.009%	0.009%	0.009%	0.008%	0.010%	0.036%
AM - %GAP (SAVEIT)	0.010%	0.164%	0.016%	0.008%	0.012%	0.036%
PVB (Index)	100	85 !!!	95	95	95	95