Imperial College London

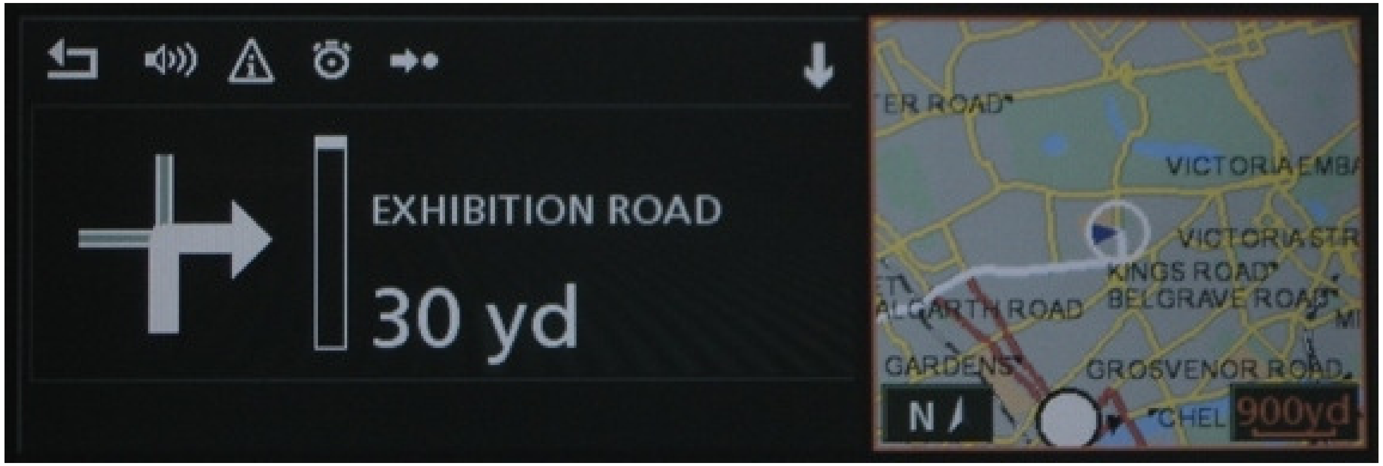
Hyperstar: A multi-path A* algorithm

7th Mathematics of Networks Meeting
Bristol University
27 June 2008

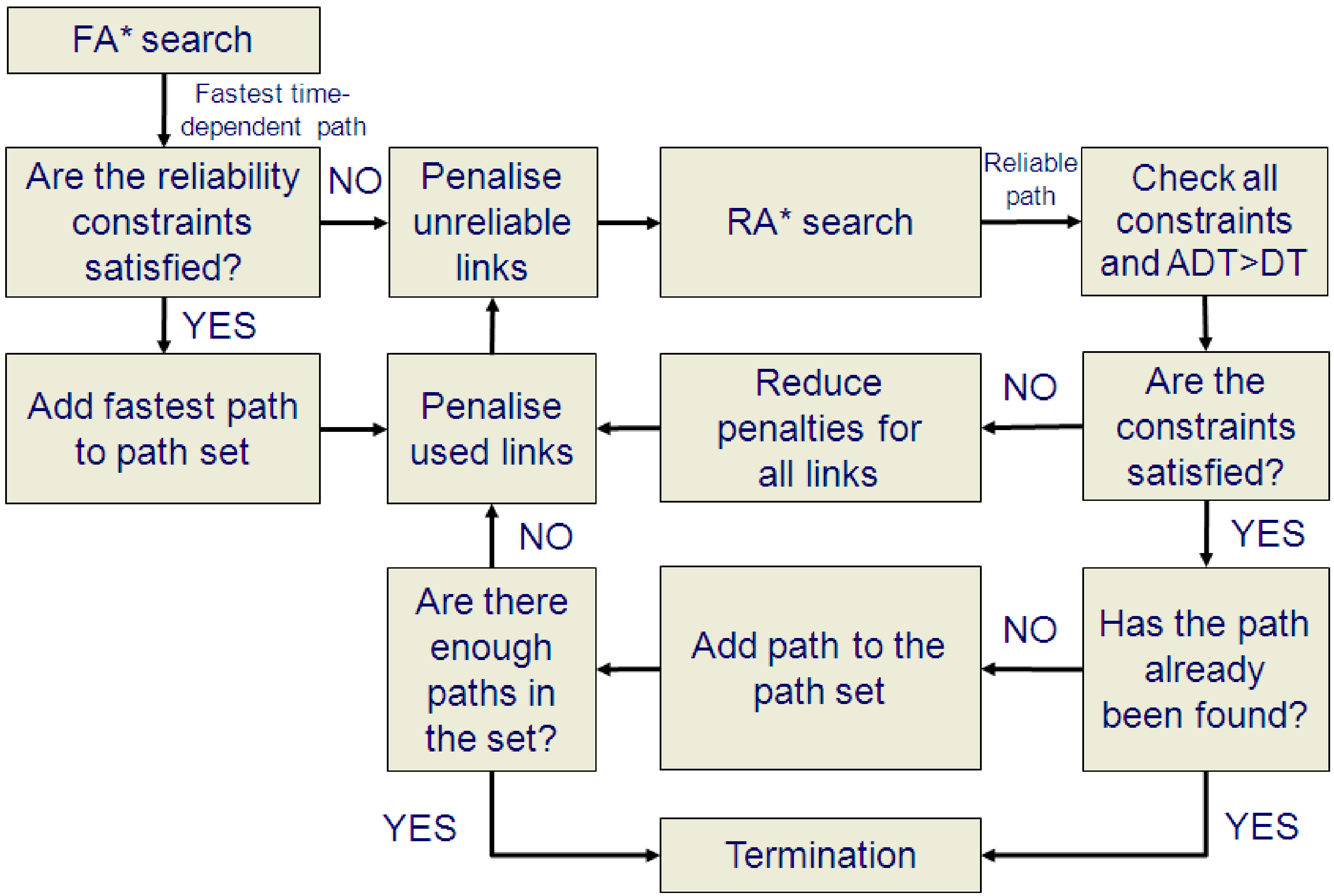
Professor Michael G H Bell
Dept of Civil & Environmental Engineering
Imperial College London

Rapid spread of Satnav

- Satnav devices have spread rapidly (4m in UK at present)
- Relatively accurate electronic maps (NavTeq, Teleatlas, etc.)
- However, link travel times are crude and seem to be based on free flow values
- TMC/TPEG congestion warning messages lead to rerouting

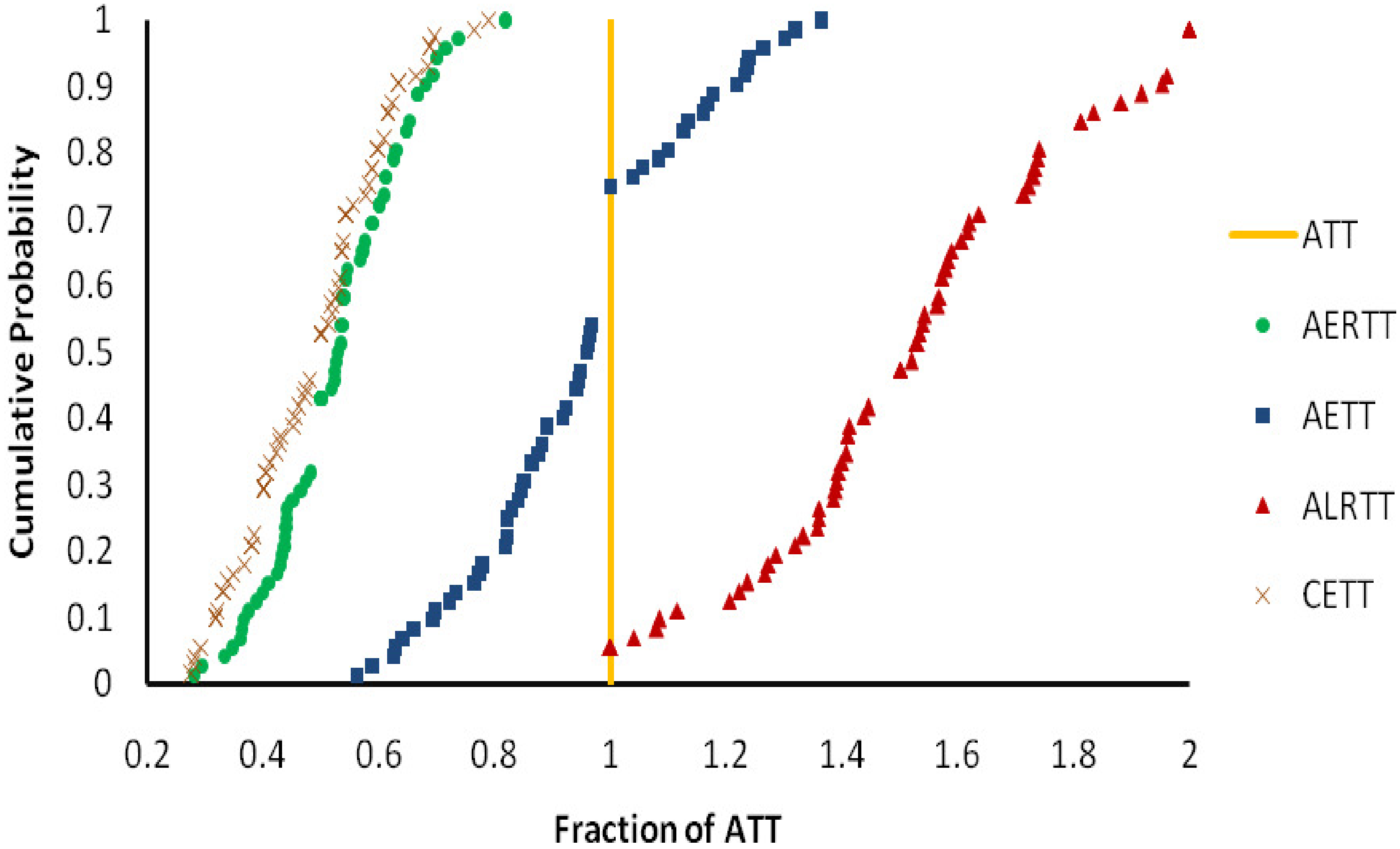


ARIAdNE: Penalty A* algorithm

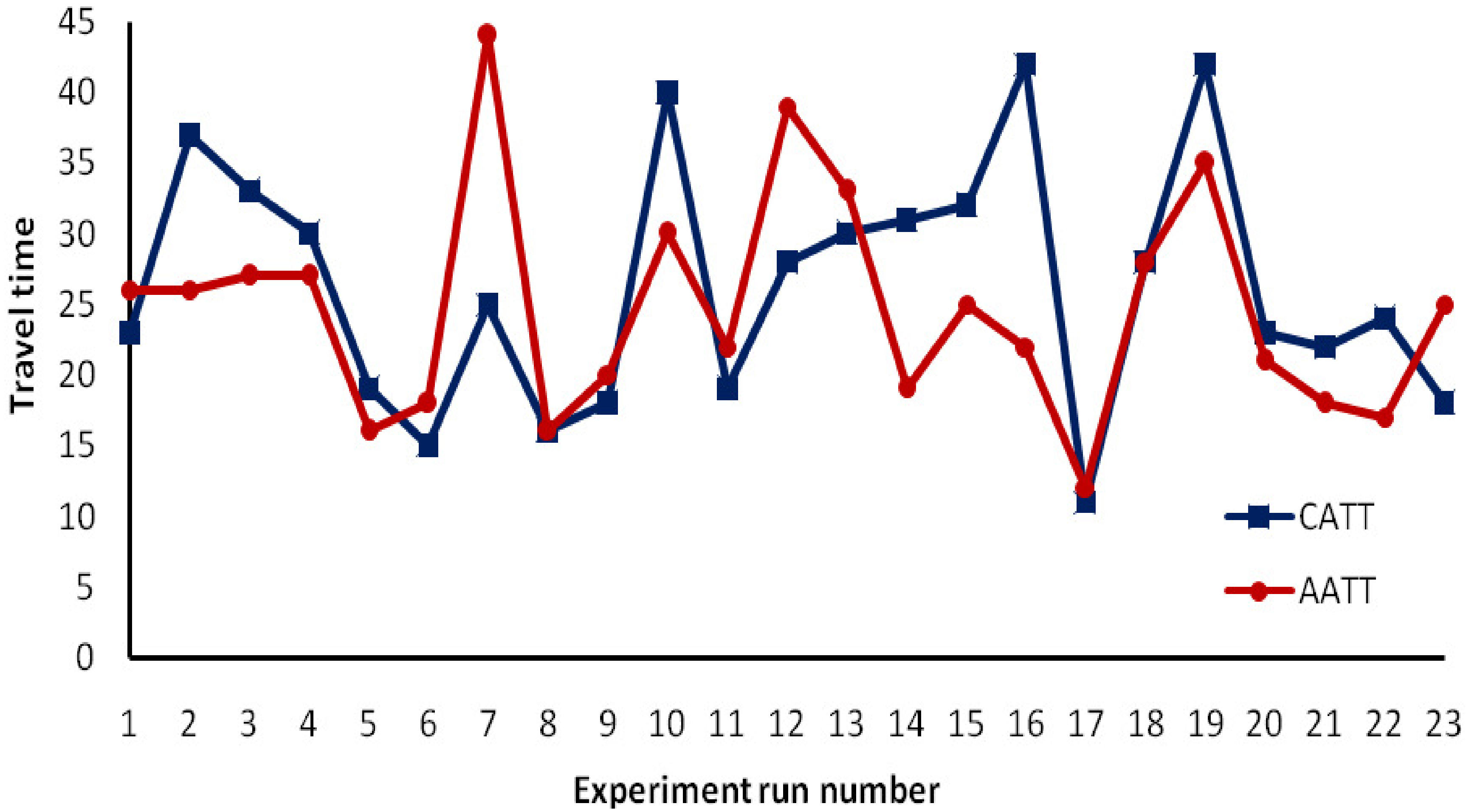


ARIAdNE field trials

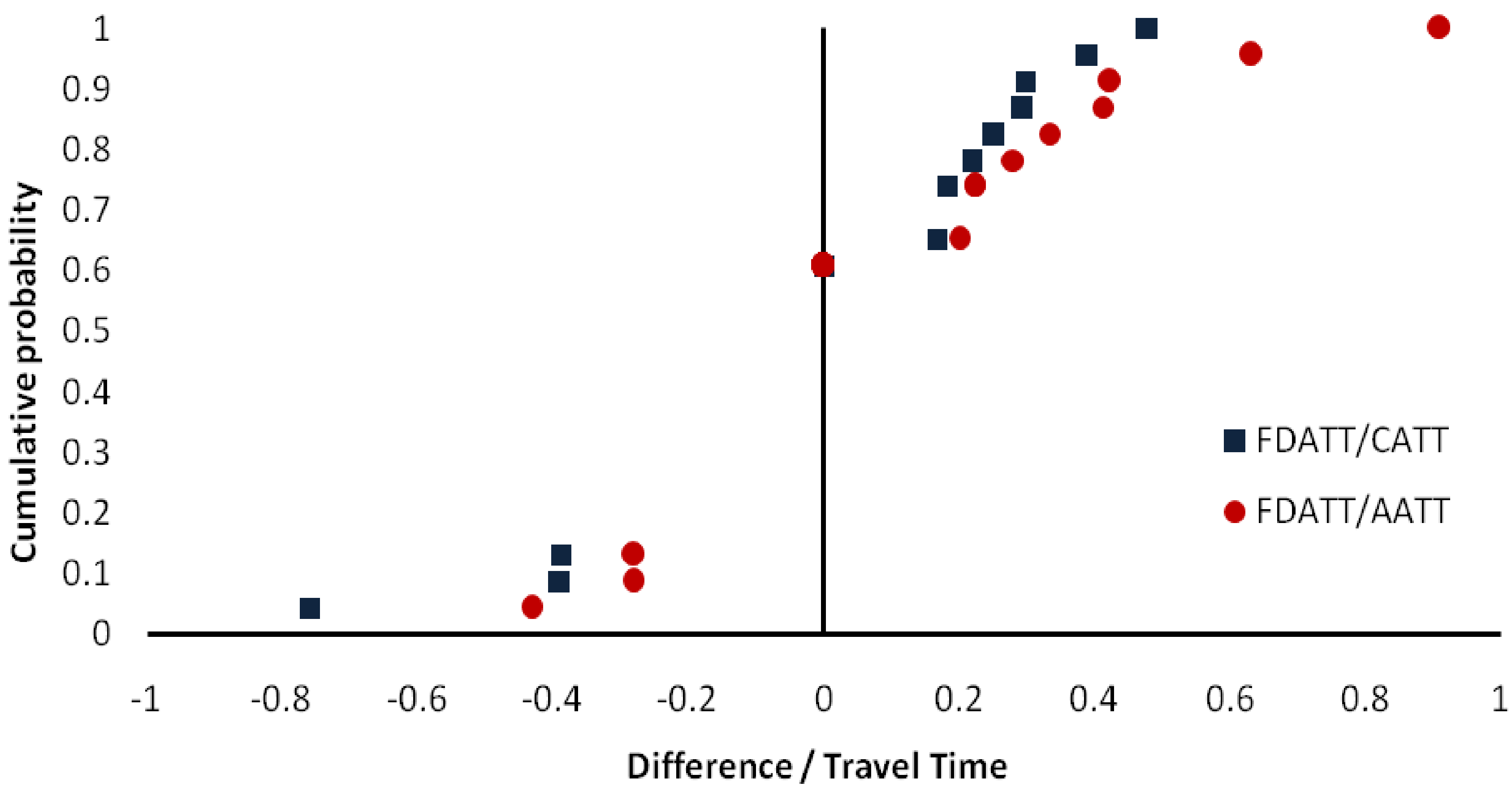
Garmin vs ARIAdNE arrival times



Garmin vs ARIAdNE routes



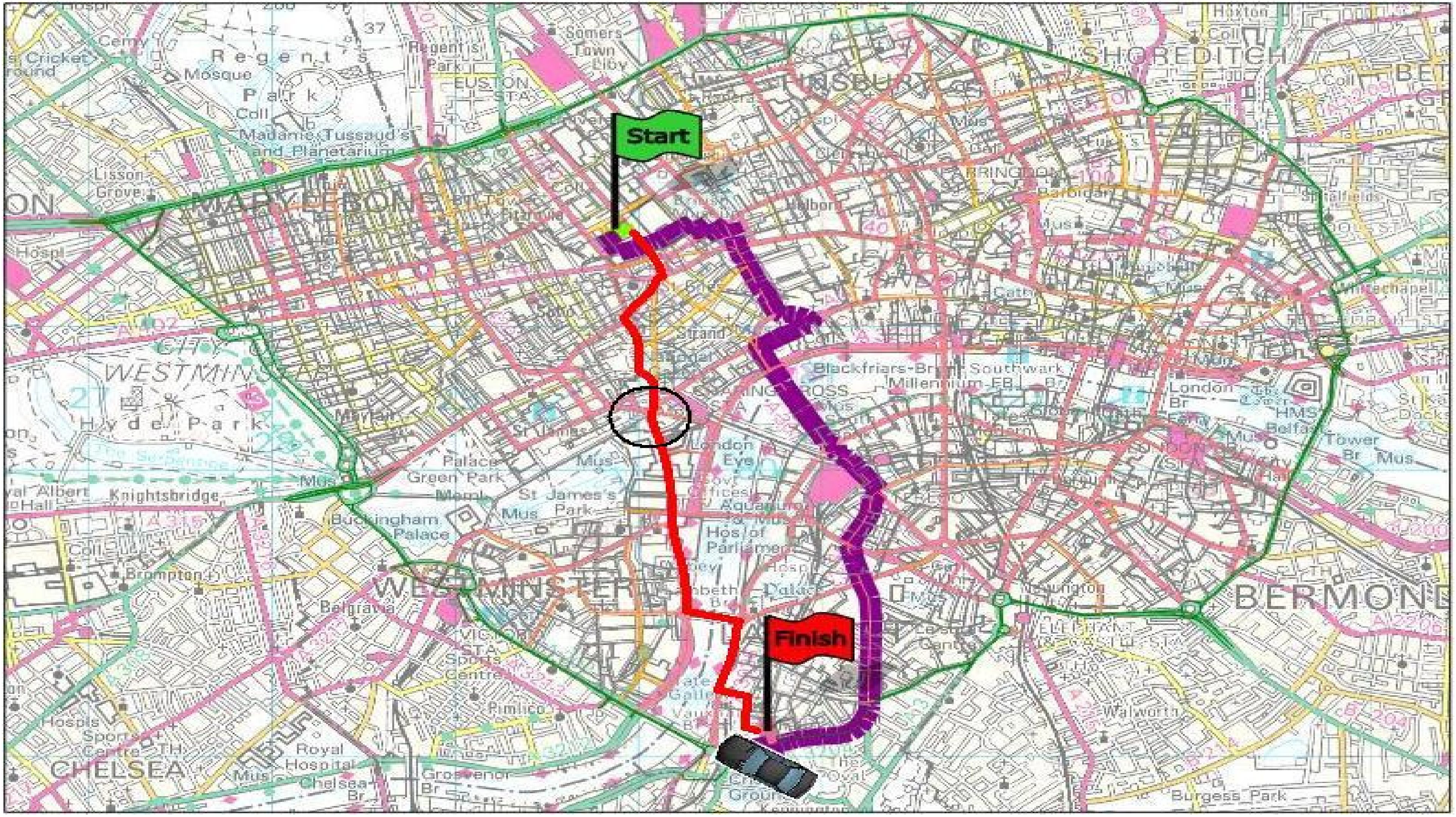
Garmin - ARIAdNE



ARIADNE worse

ARIADNE better

Why ARIAdNE works better



Comments

- The penalty A* algorithm works
- But, nice to be able to generate all routes of interest at once
- Spiess and Florian hyperpath algorithm does something similar
- Can it be adapted?

Dijkstra's algorithm

1. Start at destination and set $u_j = \infty$ for $j \neq$ destination and $u_{dest} = 0$
2. Put *dest* in OPEN
3. Search OPEN for smallest u_i
4. For nodes j reached from i if $u_j > u_i + c_{ij}$ then $u_j = u_i + c_{ij}$
5. Put nodes j in OPEN and transfer i to CLOSED
6. Return to Step 3 until origin in CLOSED

A* algorithm

1. Start at destination and set $u_j = \infty$ for $j \neq$ destination and $u_{dest} = 0$
2. Put *dest* in OPEN
3. Search OPEN for smallest $u_i + h_{i,orig}$
4. For nodes j reached from i if $u_j > u_i + c_{ij}$ then $u_j = u_i + c_{ij}$
5. Put nodes j in OPEN and transfer i to CLOSED
6. Return to Step 3 until origin is CLOSED

Hyperpath algorithm

- *Hyperpath* is a bundle of potentially optimal paths
- Every link has both a cost and a service frequency
- Where there is choice within the hyperpath, allocation is proportional to service frequency (the *strategy*)
- Elemental path only added to hyperpath if the expected cost of travel is reduced

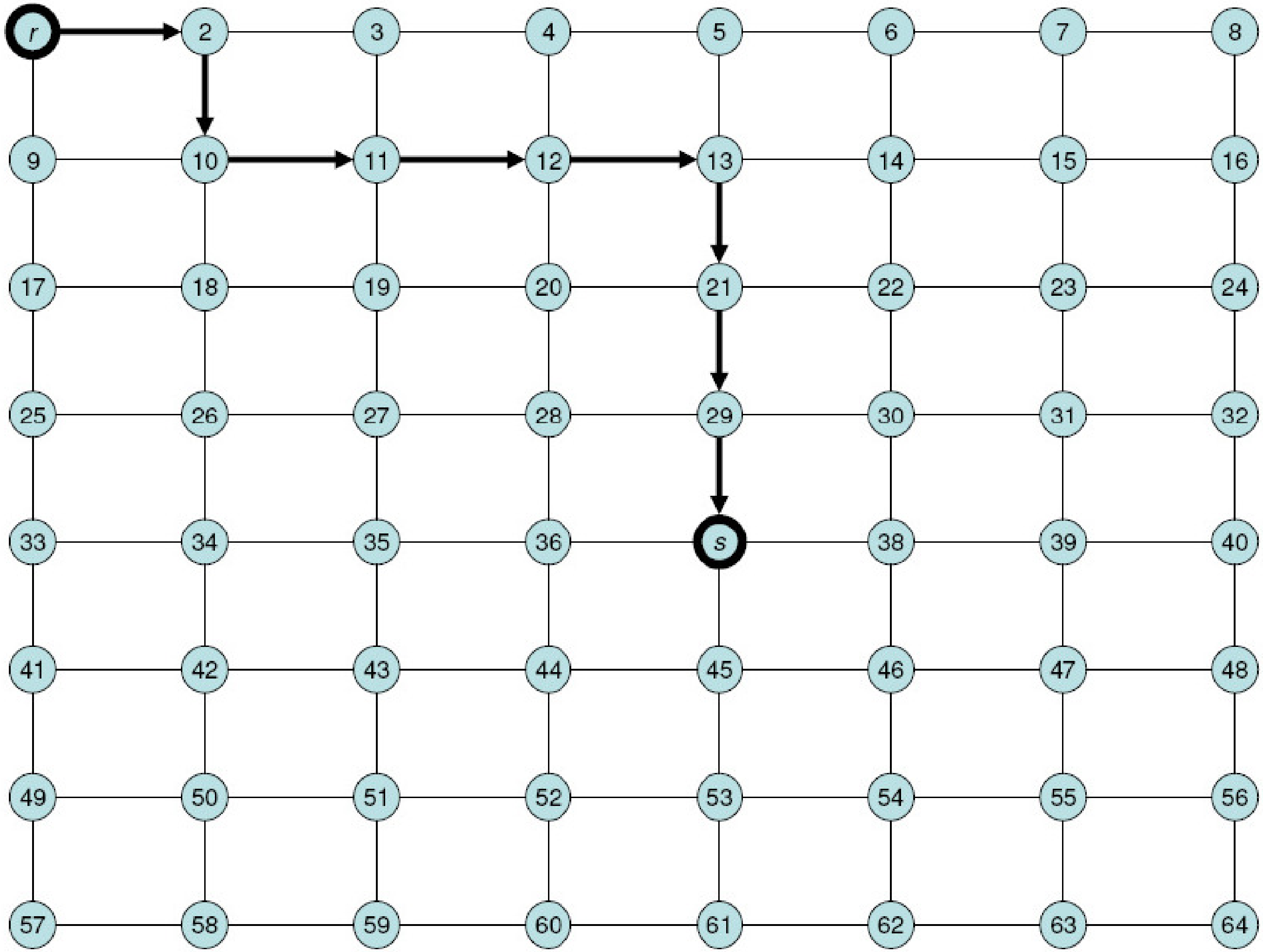
Hyperpath algorithm

1. Start at destination and set $u_j = \infty$ for $j \neq$ destination, $u_{dest} = 0$ and $F_i = 0$
2. Put *dest* in OPEN
3. Search OPEN for smallest u_i
4. For nodes j reached from i if $u_j > u_i + c_{ij}$ then $u_j = (F_i\, u_i + f_{ij}\, c_{ij}) / (F_i + f_{ij})$, $F_i = F_i + f_{ij}$ and add link (i,j) to HYPERPATH
5. Put nodes j in OPEN and transfer i to CLOSED
6. Return to Step 3 until origin is CLOSED

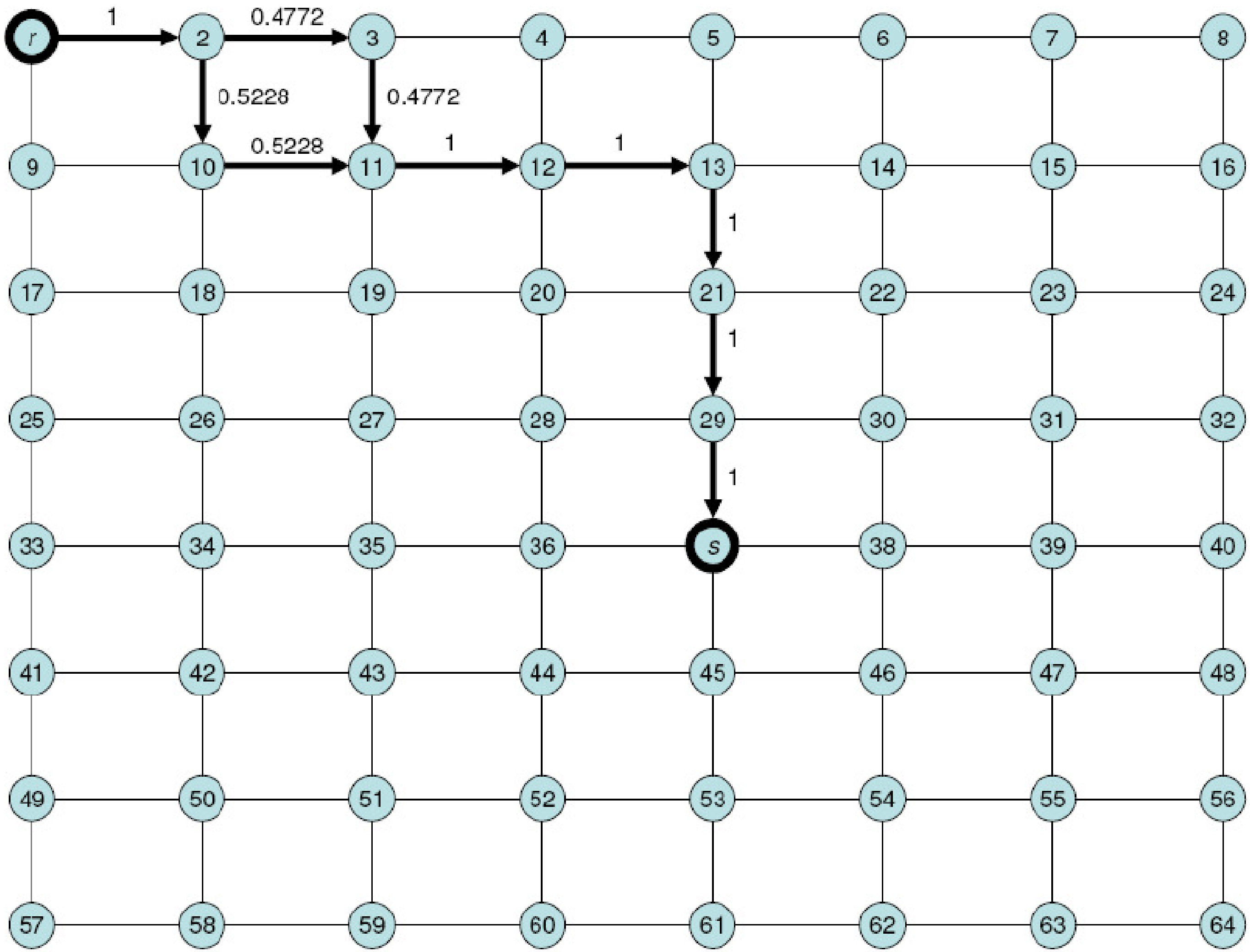
Reinterpreting the hyperpath algorithm

- Note: $1 / f_{ij}$ = link headway = max link delay = d_{ij}
- Allocation: Minmax exposure to delay
  - $\Rightarrow p_{ij}\, d_{ij} = p_{ik}\, d_{ik}$ if links (i,j) and (i,k) attractive
  - $\Rightarrow p_{ij} \propto 1 / d_{ij} = f_{ij}$
- Attractive: Add link to hyperpath if "expected" travel time reduced. Expected by whom? A risk averse traveller.

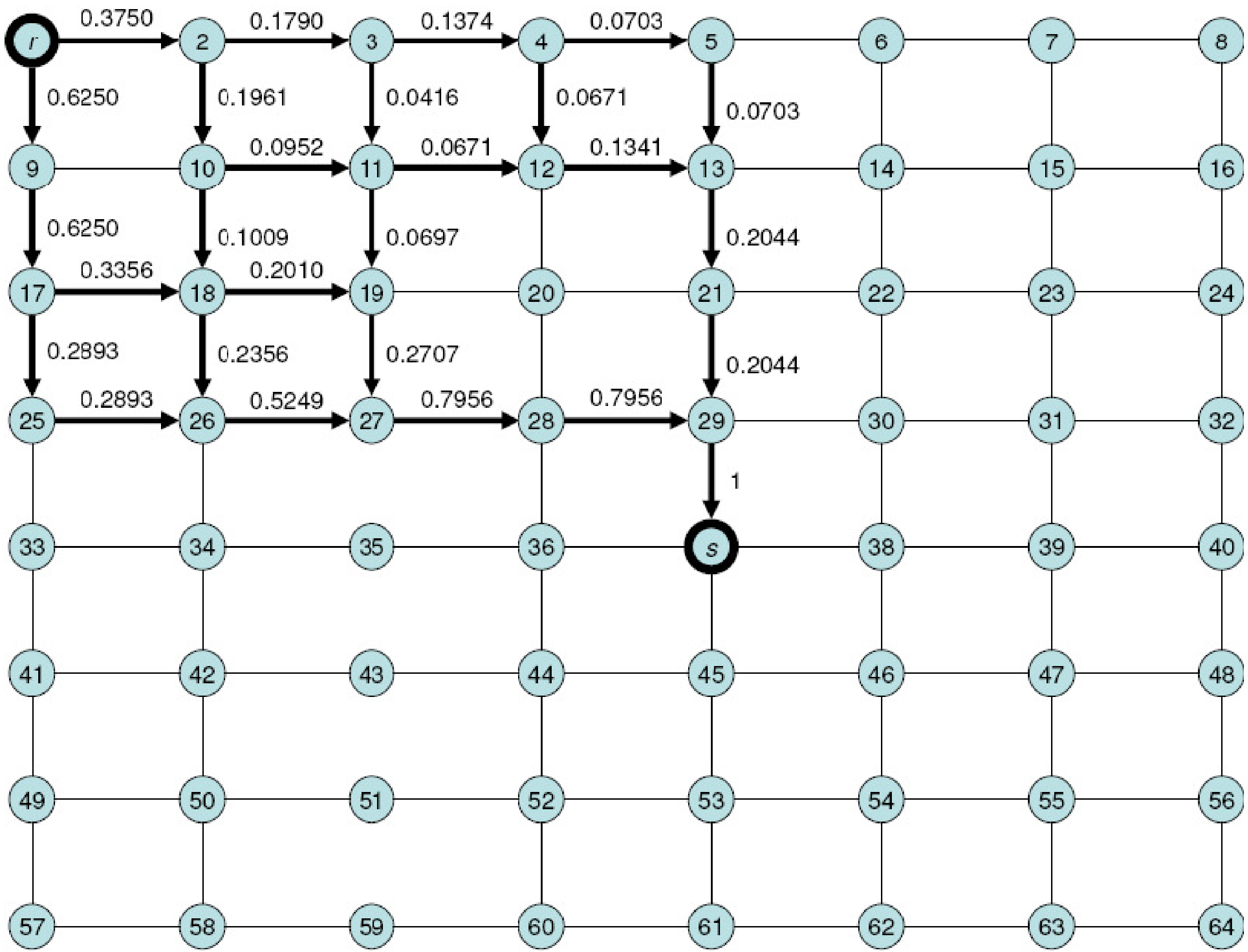
Singular hyperpath: No delay



Hyperpath: Medium max link delays



Hyperpath: Large max link delays



Effect of A* speed-up

Table 2: Comparative performance of the Hyperstar algorithm

	Maximum delay	u_r	A_0 selected links	A_1 selected links
Case 1	$d = 0$	10.7001	219	79
Case 2	$d = 0.3R$	11.8649	222	111
Case 3	$d = R$	13.6226	223	148

H* algorithm

1. Start at destination and set $u_j = \infty$ for $j \neq$ destination, $u_{dest} = 0$ and $F_i = 0$
2. Put *dest* in OPEN
3. Search OPEN for smallest $u_i + h_{i,orig}$
4. For nodes j reached from i if $u_j > u_i + c_{ij}$ then $u_j = (F_i\, u_i + f_{ij}\, c_{ij}) / (F_i + f_{ij})$, $F_i = F_i + f_{ij}$ and add link (i,j) to HYPERPATH
5. Put nodes j in OPEN and transfer i to CLOSED
6. Return to Step 3 until origin is CLOSED

Discussion

- Approaches for handling uncertain delays in road networks examined in context of vehicle navigation $\Rightarrow$ must be efficient $\Rightarrow$ based on A*
- Approach 1: Avoid unreliable links $\Rightarrow$ Penalty A* method
- Approach 2: Seek bundle of routes that may be optimal by adapting A*, with actual route determined by TMC/TPEG messages

Danke für Ihre Aufmerksamkeit!

Fragen?