Preferred citation style for this presentation

M. Balmer (2007)
Fast shortest path computation in time-dependent traffic networks, paper presented at the 2th MATSim Seminar, Castasegna, Oktober 2007.

Fast shortest path computation in time-dependent traffic networks

M. Balmer

IVT
ETH
Zurich

MATSim Seminar Oktober 2007

Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

MATSim-T: Routing in agent based micro-simulations (2)

Task: Finding the time dependent least cost path from O to D
in a dynamically weighted, strongly connected di-graph.
➔ Complexity: $O(n^2)$

Street networks are sparse graphs.
➔ Complexity: $<O(n^2)$

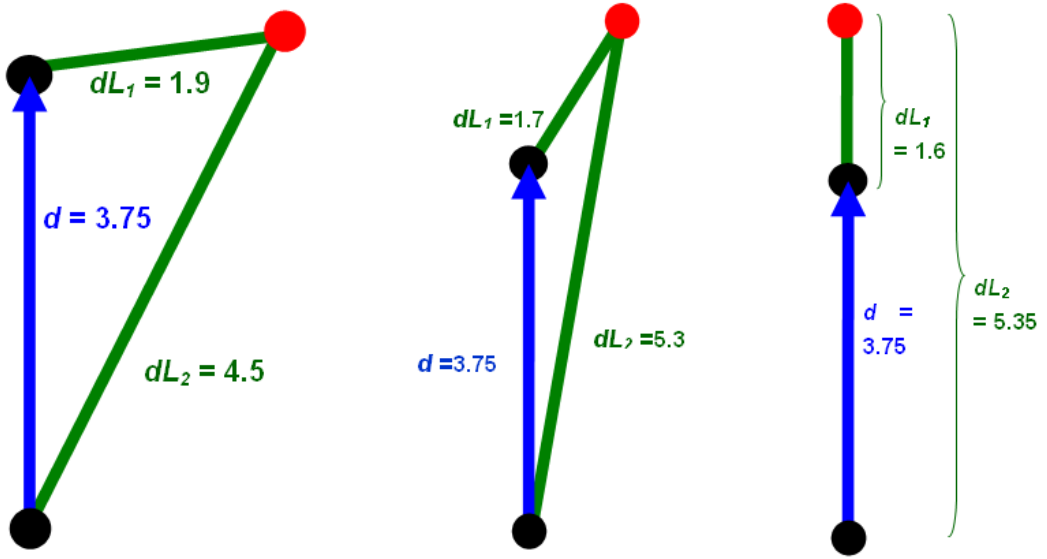
Number of routes to calculate (Example: Switzerland)
- ca. 7.5 Mio. inhabitants
- ca. 3 trips per person
- ca. 100 iterations with 10% re-routing (MATSim-T)

➔ ca. 225 Mio. routes to calculate

Landmarks A*-Algorithm (1)

A*-Algorithm:
Estimation of remaining cost ($h_D(x)$) during the route finding process.
Estimated cost at x: $c_x(O)+h_D(x)$
➔ Rule: $h_D(x)$ <= remaining cost ($c_D(x)$)

Landmarks A*-Algorithm:
$h_D(x)$ estimation via triangle inequality



5.5 – 1.9 = 2.6 < 3.75

5.3 – 1.7 = 3.6 < 3.75

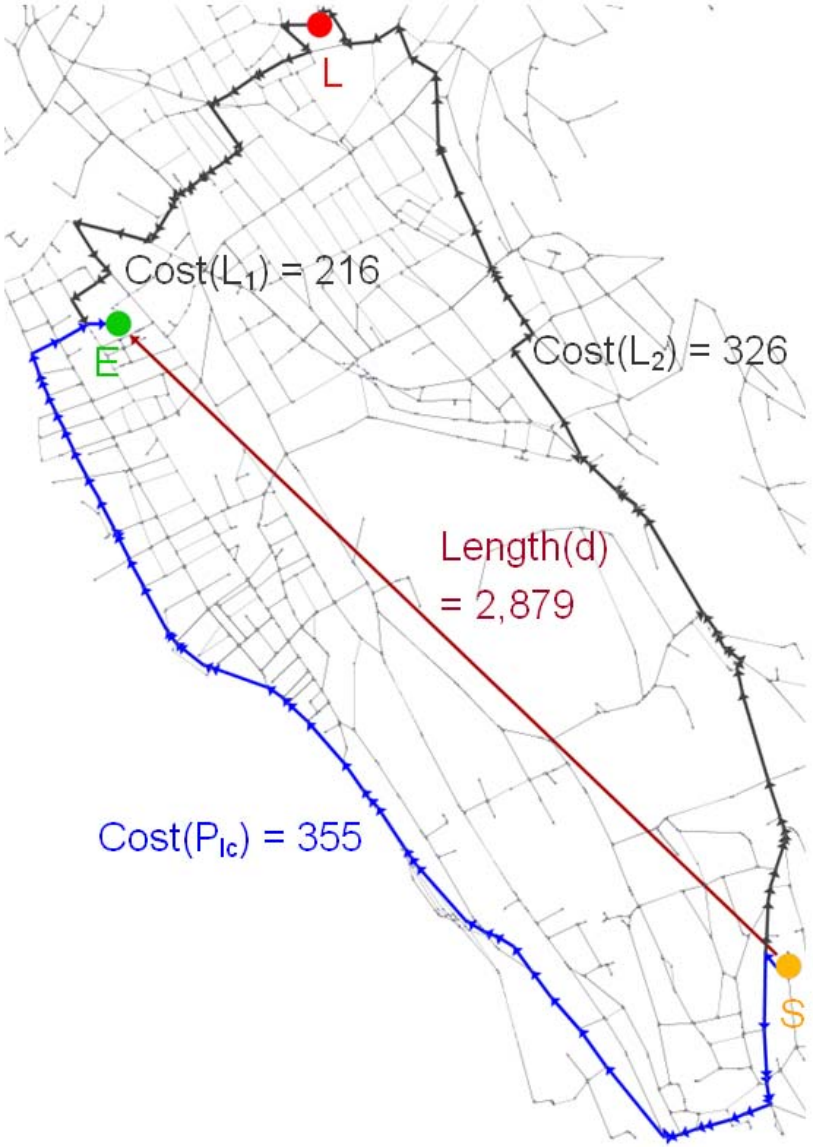
5.35 – 1.6 = 3.75

Landmarks A*-Algorithm (2)

Problems in travel time estimation in traffic networks:
- different free speed per link
- dynamic link travel time

To guarantee $h_D(x) <= c_D(x)$:
Cost(L_i) calculated on an unloaded network

Pre-process:
calculate and store Cost(L_i) from each node of the network.



Test Sets: Networks

Name	# nodes	# links	Avg. node degree	Detour Index	Avg. link length (m)	Avg. link free speed (m/s)	# dead-end nodes
Net1	7,377	20,252	5.49	1	1,816.82	14.20	1,025
Net2	14,803	39,372	5.32	0.90	1,218.43	19.40	1,776
Net3	408,636	882,120	4.32	0.97	96.68	11.00	127,904
Net4	471,879	1,053,259	4.47	0.95	108.49	9.29	142,238

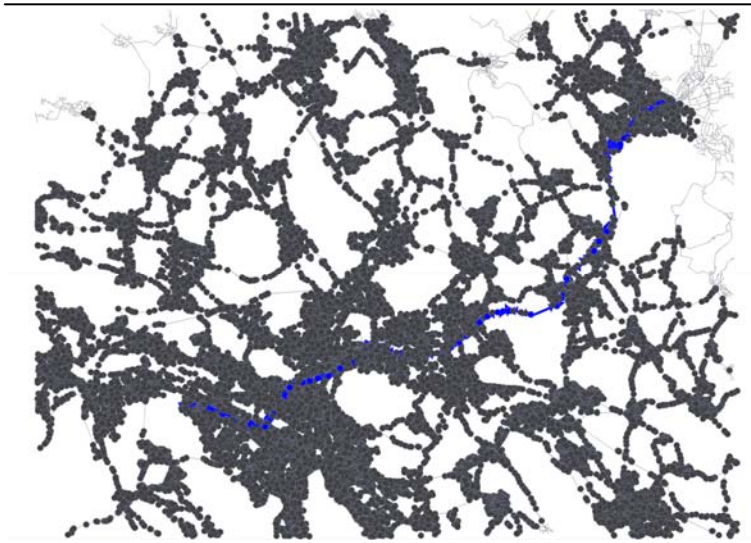
Test Sets: Routing Algorithms

- Basic Dijkstra (Dijk)
- Iteration-ID Dijkstra (It-ID Dijk)
- Iteration-ID Dijkstra with dead-end pruning (Prune Dijk)
- Euclidean A* (Euc A*)
- Landmarks A* (Landm A*)
- Overdo A* (Ov A* X, where X denotes the overdo factor)

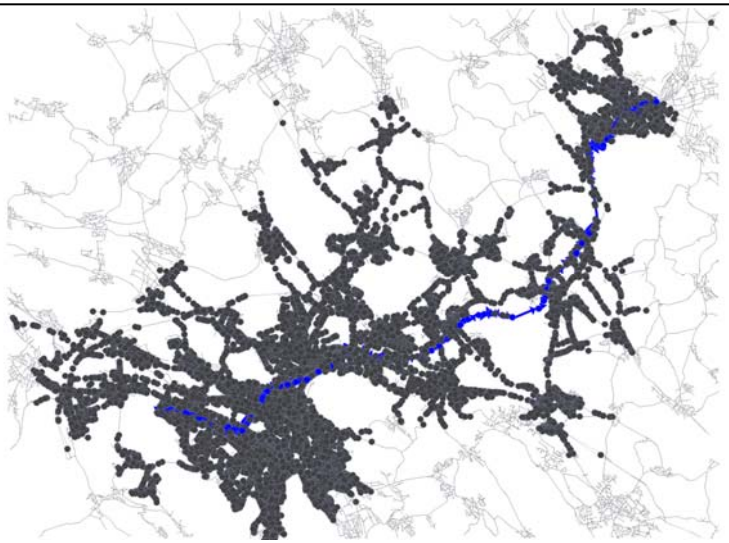
Test Sets: Demand (Trips)

Aimed from-to distance	Avg. from-to distance Net1	Avg. from-to distance Net2	Avg. from-to distance Net3	Avg. from-to distance Net4
1 km	1.86 km	1.68 km	1.17 km	1.17 km
5 km	5.27 km	5.38 km	5.06 km	5.06 km
25 km	25.29 km	25.21 km	25.06 km	25.06 km
50 km	49.98 km	50.00 km	50.02 km	50.02 km
100 km	99.83 km	100.01 km	100.02 km	100.02 km
150 km	149.67 km	150.20 km	150.03 km	150.03 km
200 km	200.36 km	200.20 km	200.04 km	200.04 km

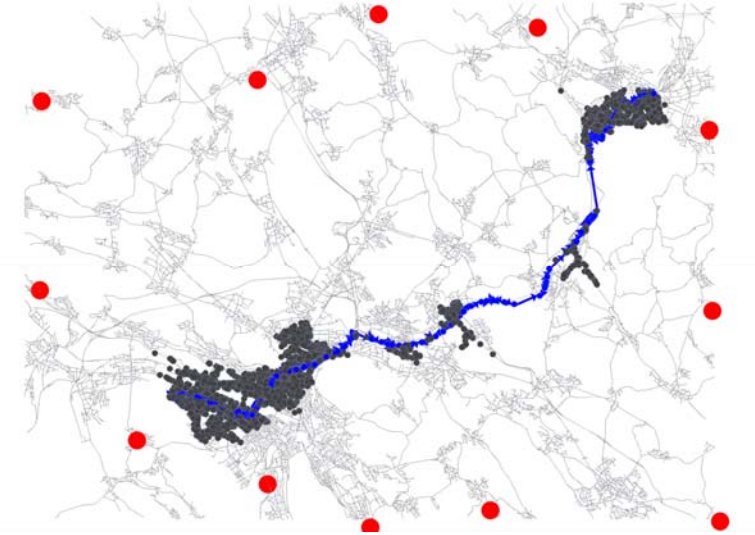
Performance



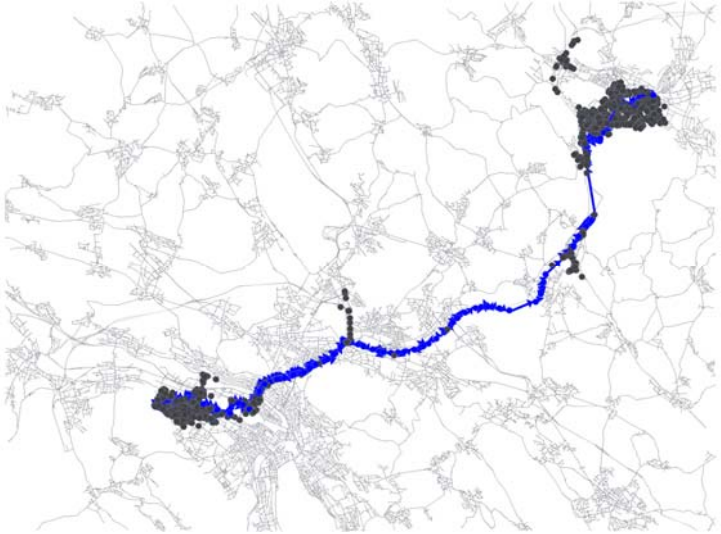
(It-ID) Dijk



Euc A*



Landm A* (12 L)

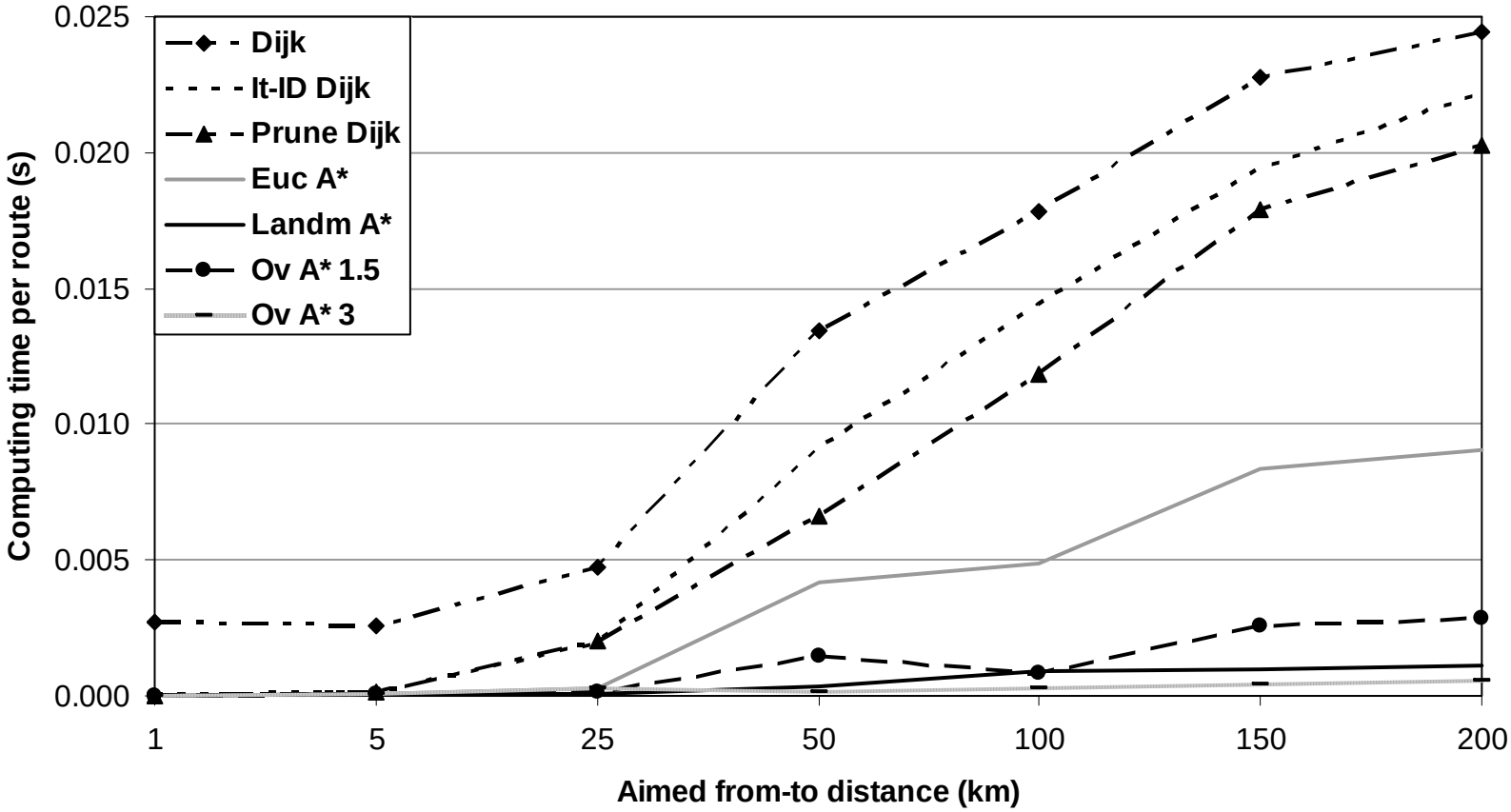


Ov A* 2

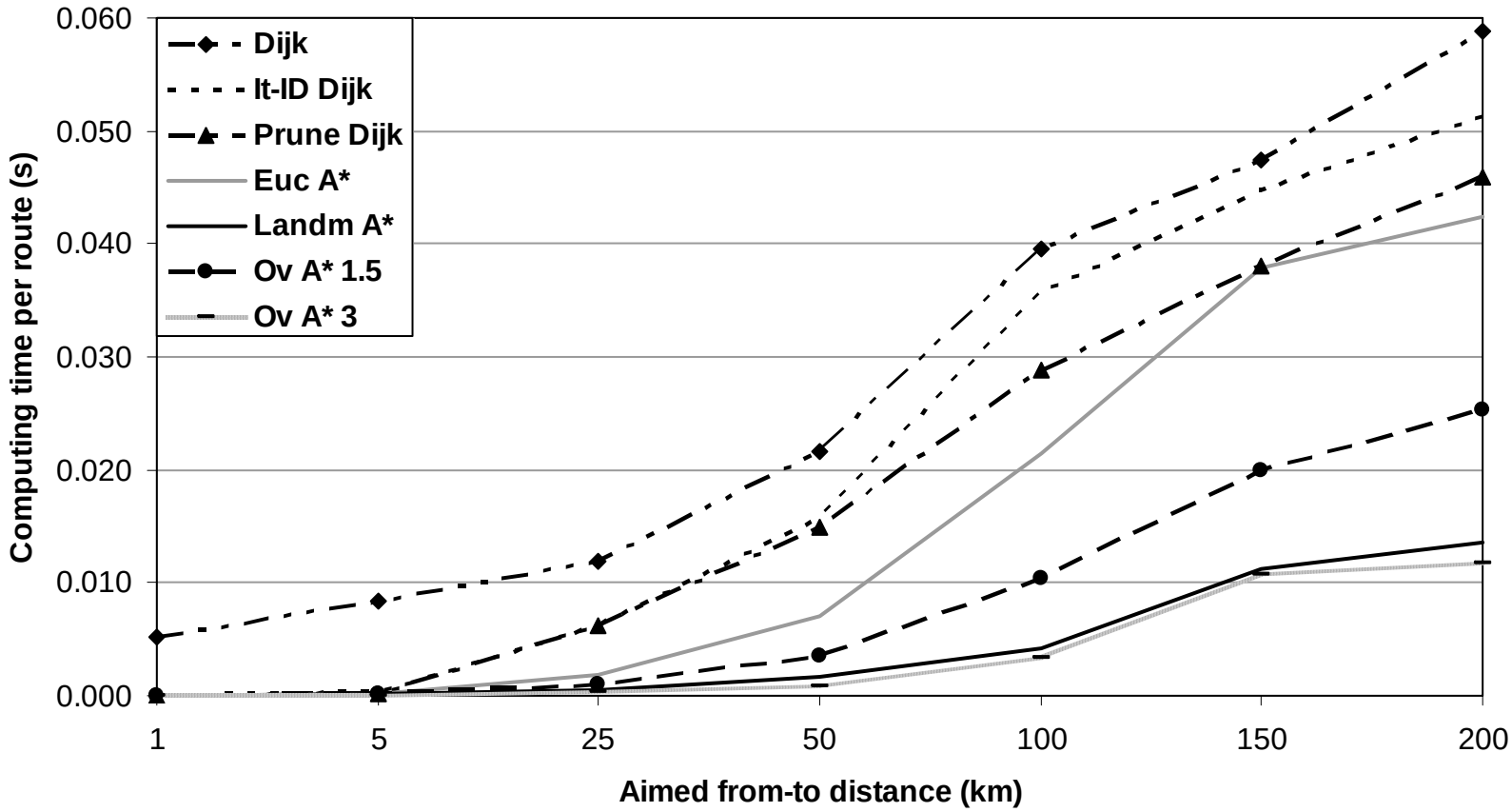
Performance: Pre-process

Algorithm	**Net1**	**Net2**	**Net3**	**Net4**
Dijk/It-ID Dijk	-	-	-	-
Prune Dijk/Euc A*/Ov A*	0.13 s	0.22 s	4.20 s	5.00 s
Landm A*	1.20 s	2.00 s	72.00 s	160.00 s

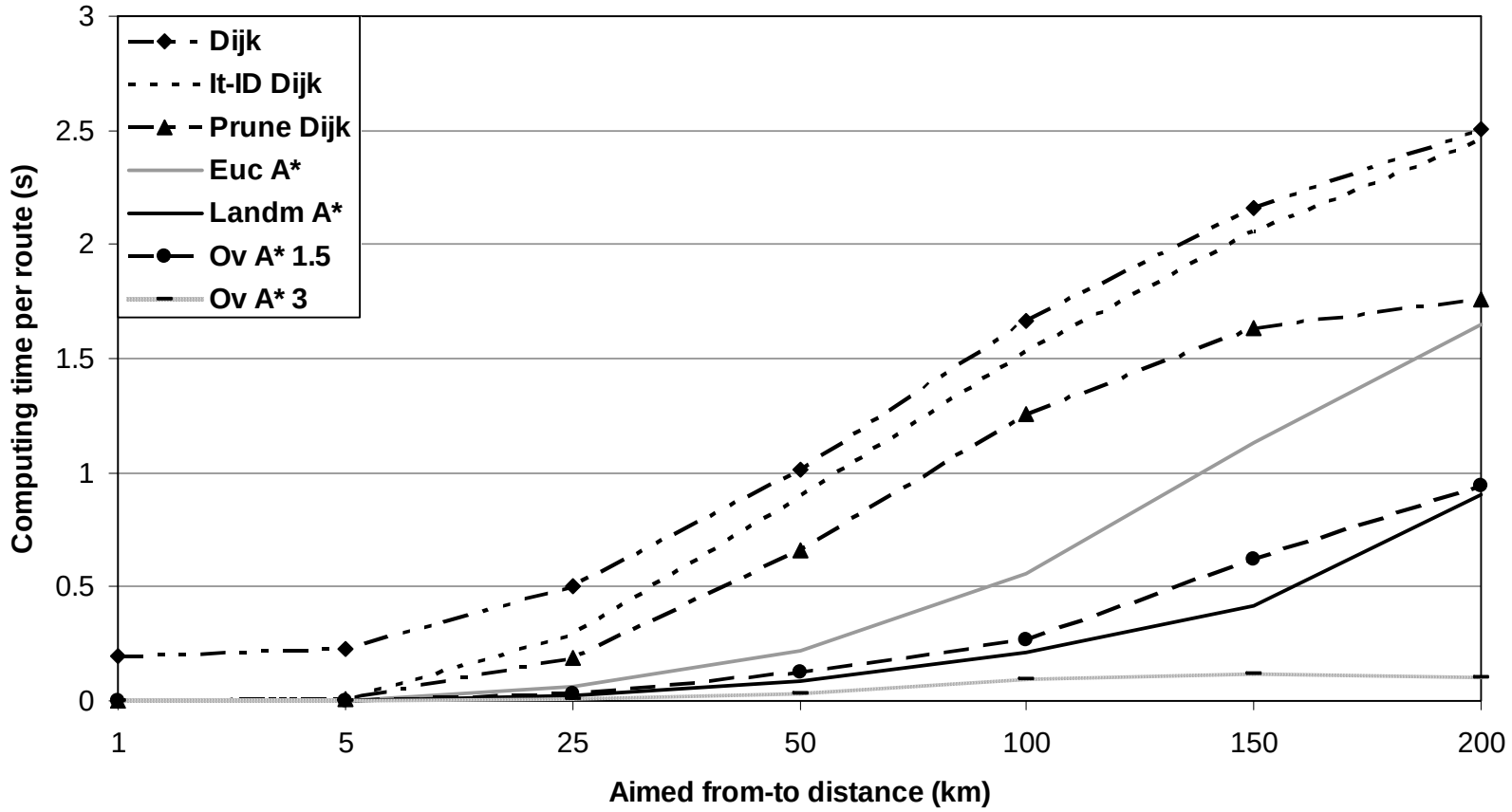
Performance: Free Flow (Net 1)



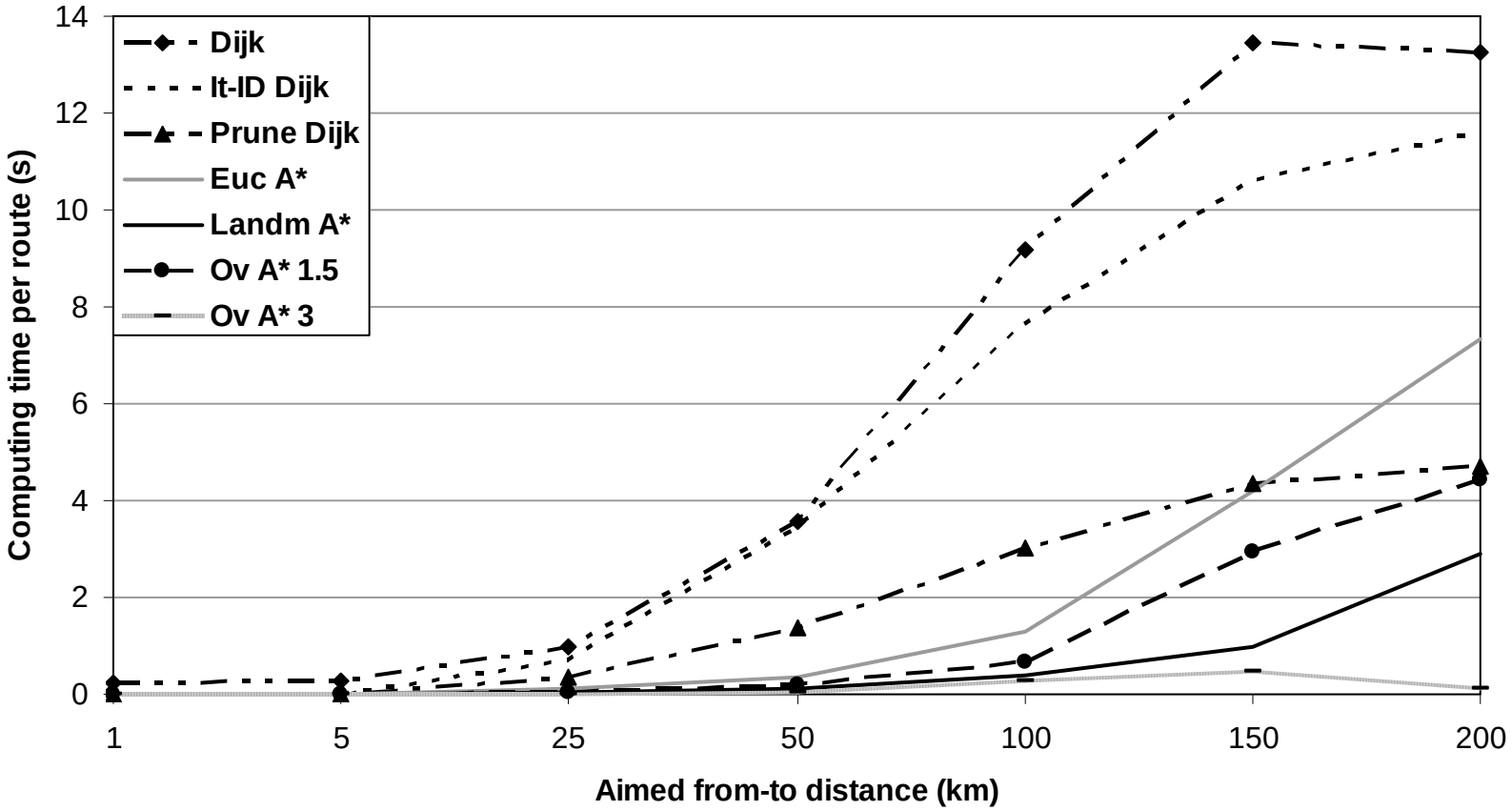
Performance: Free Flow (Net 2)



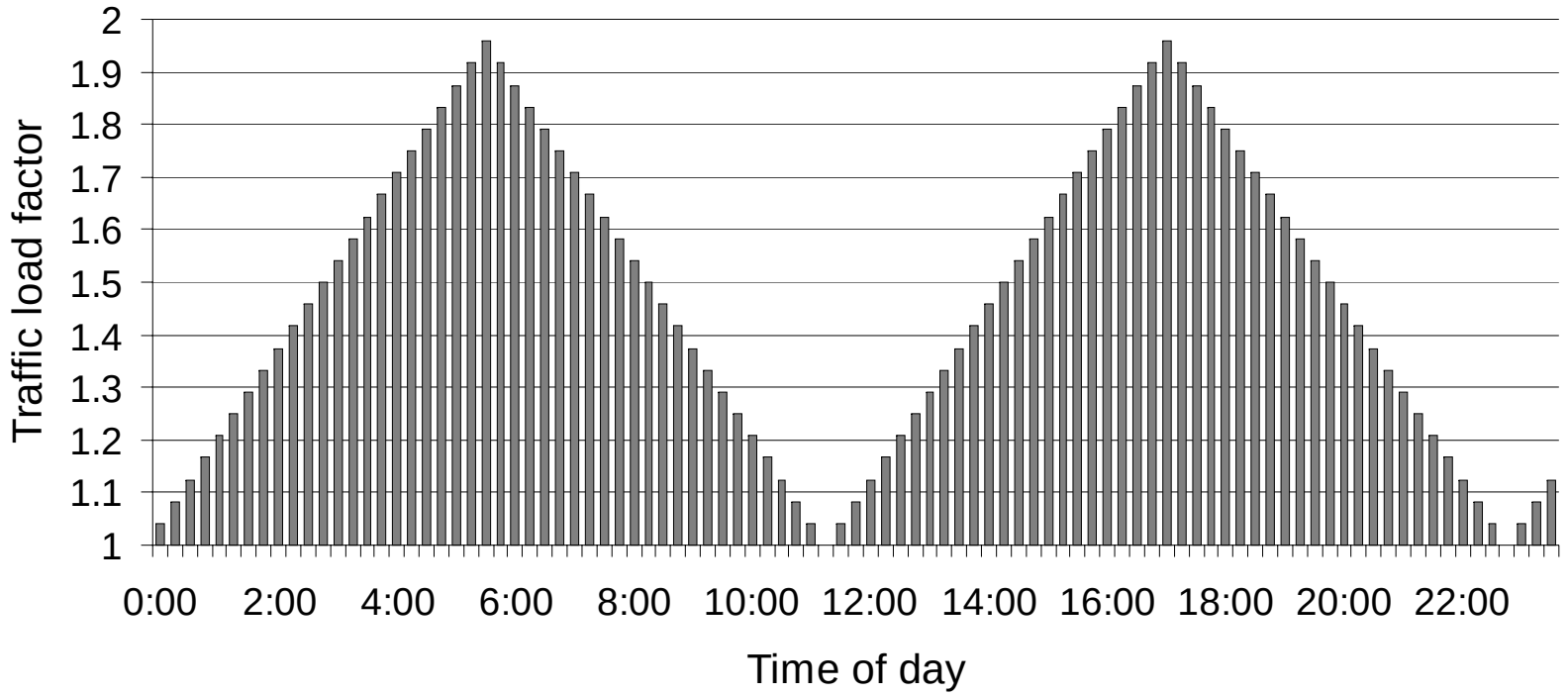
Performance: Free Flow (Net 3)



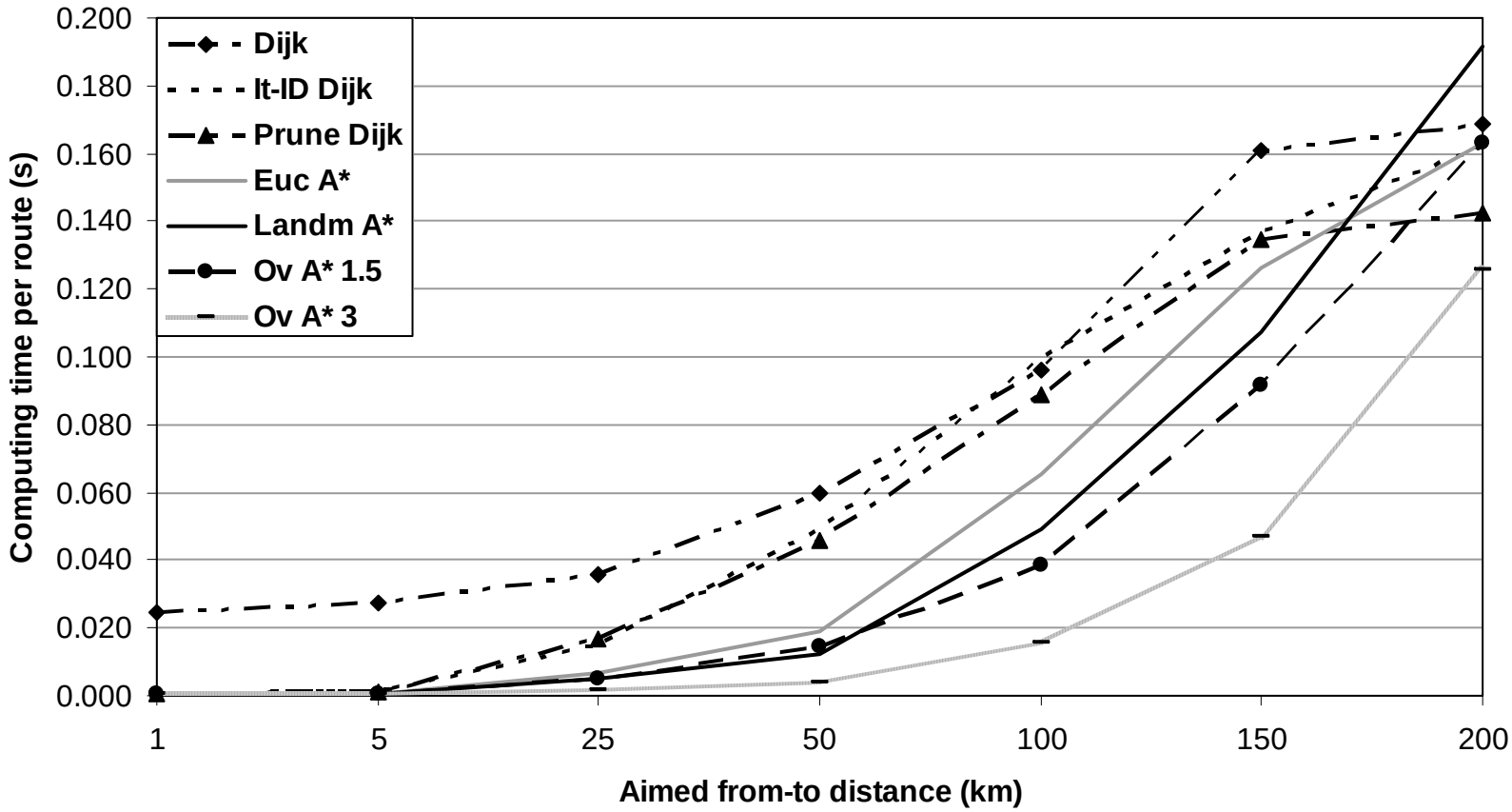
Performance: Free Flow (Net 4)



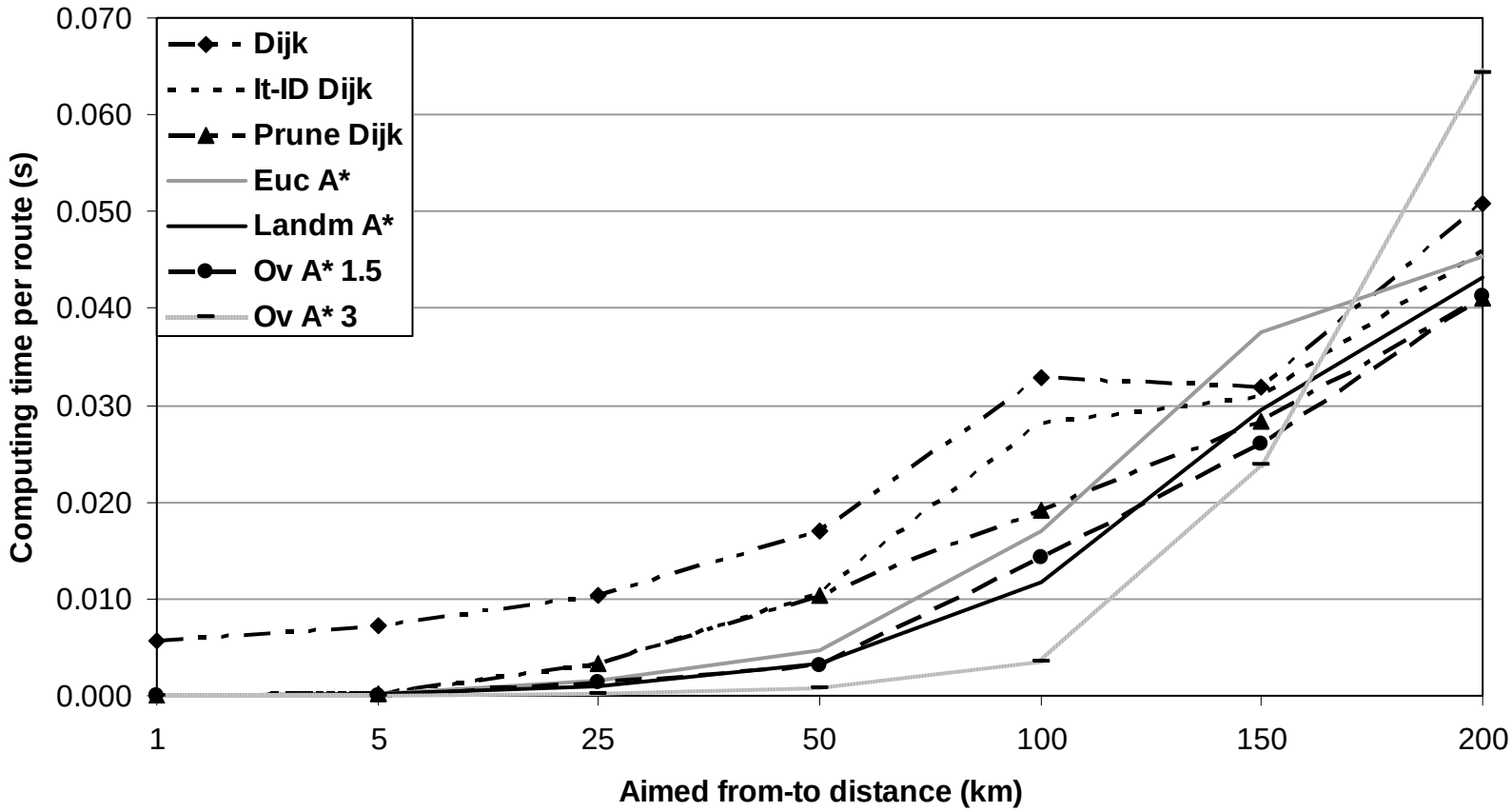
Performance: Loaded (synthetic)



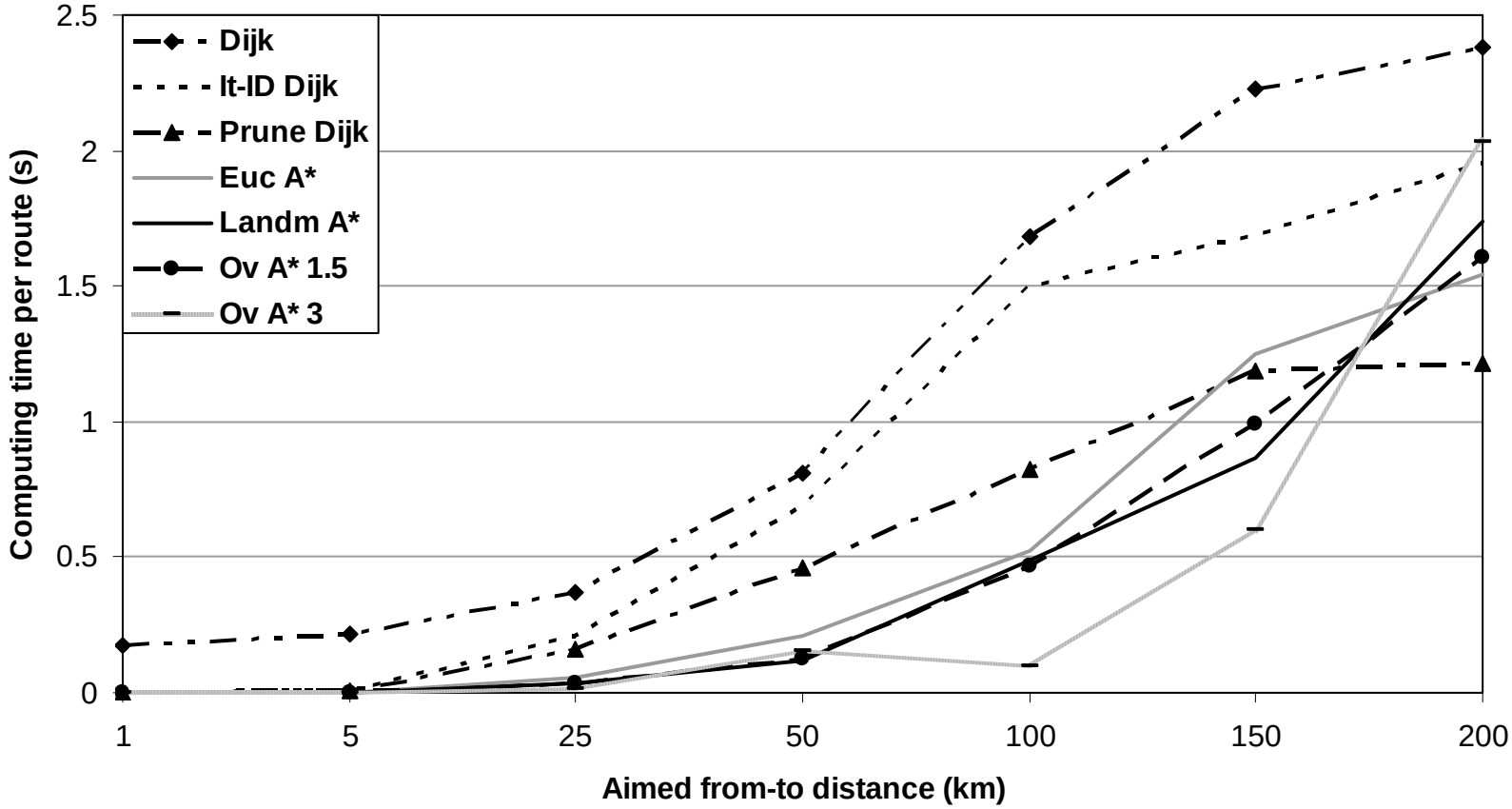
Performance: Loaded (Net 1)



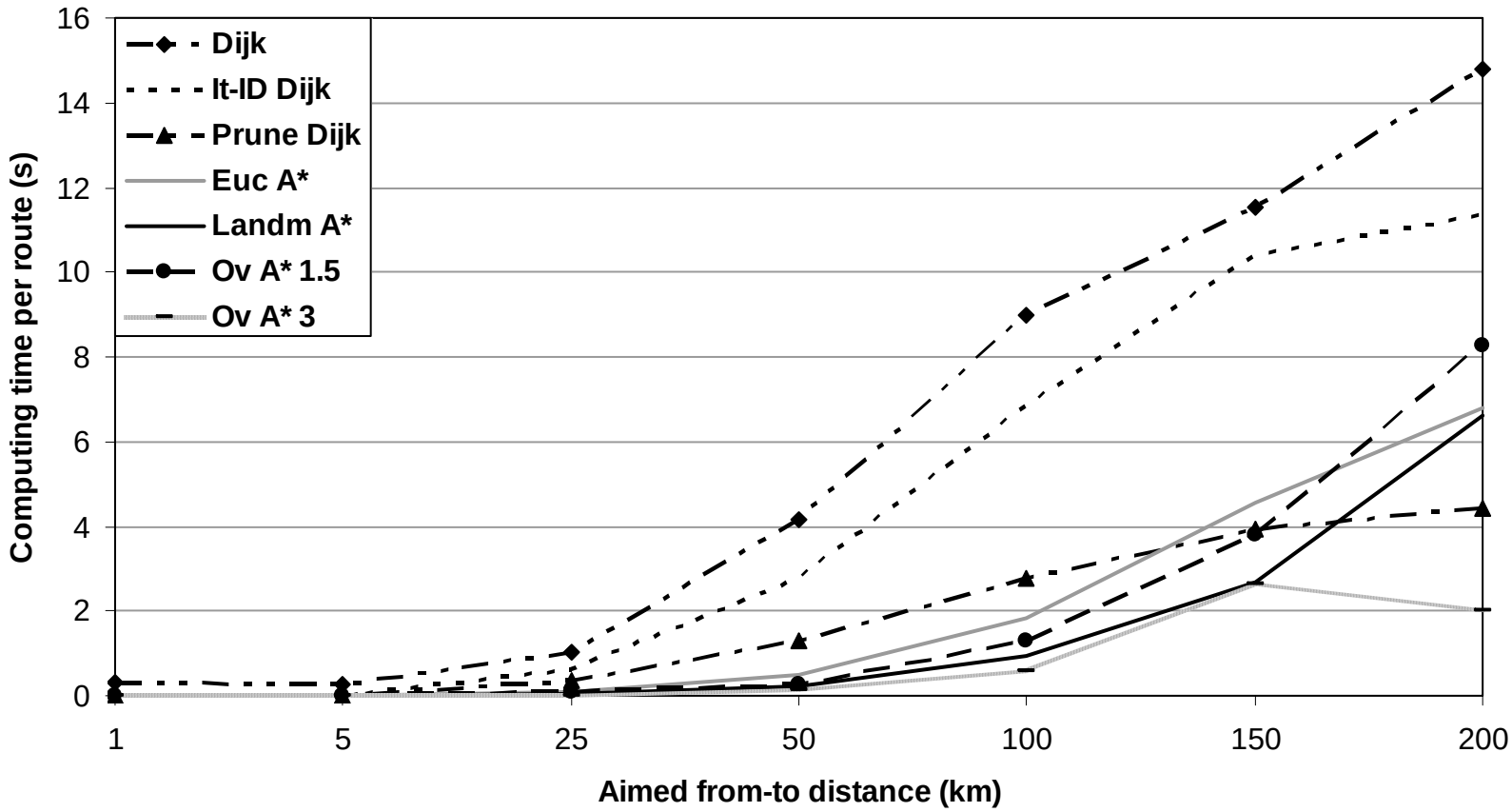
Performance: Loaded (Net 2)



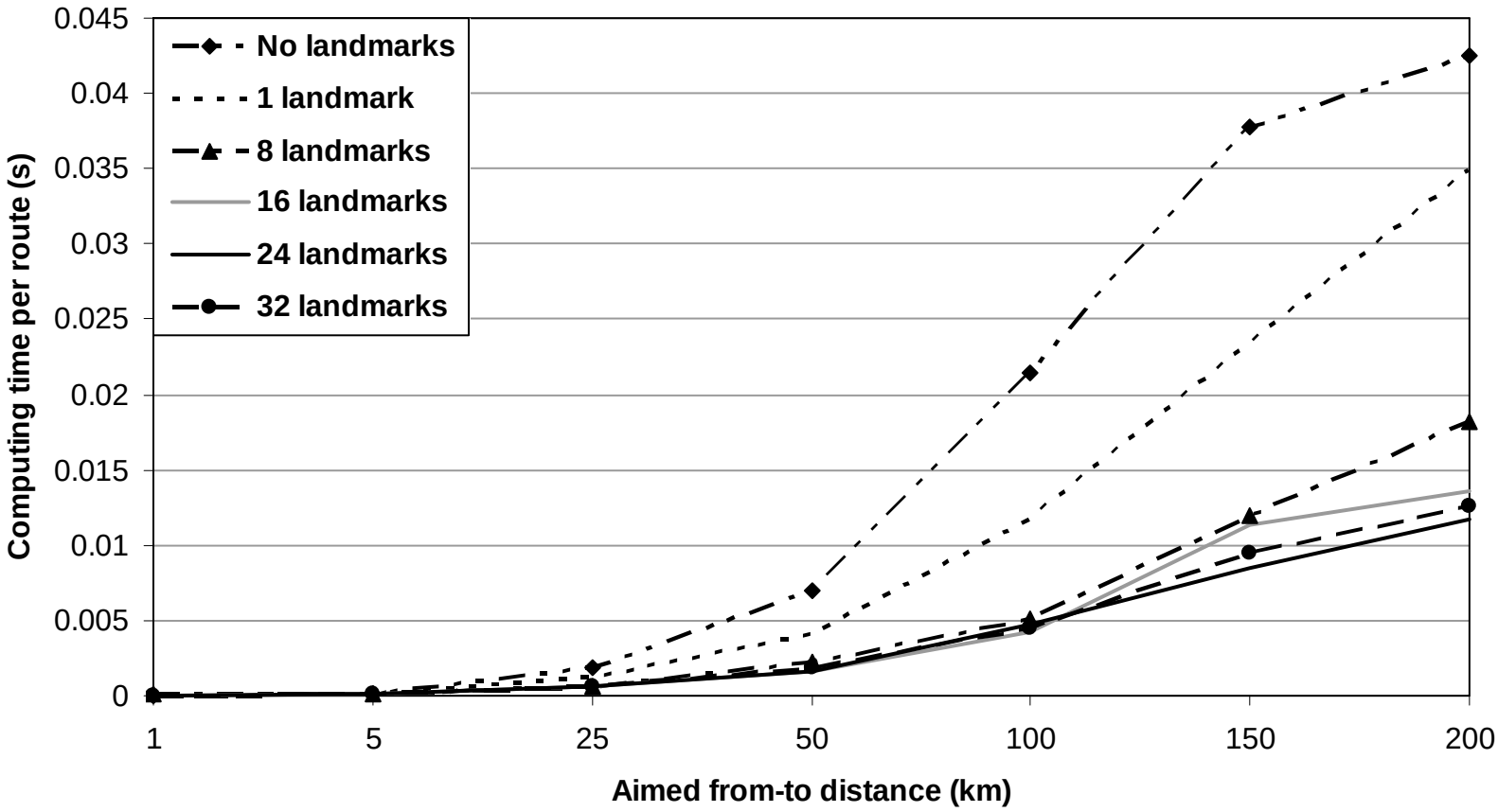
Performance: Loaded (Net 3)



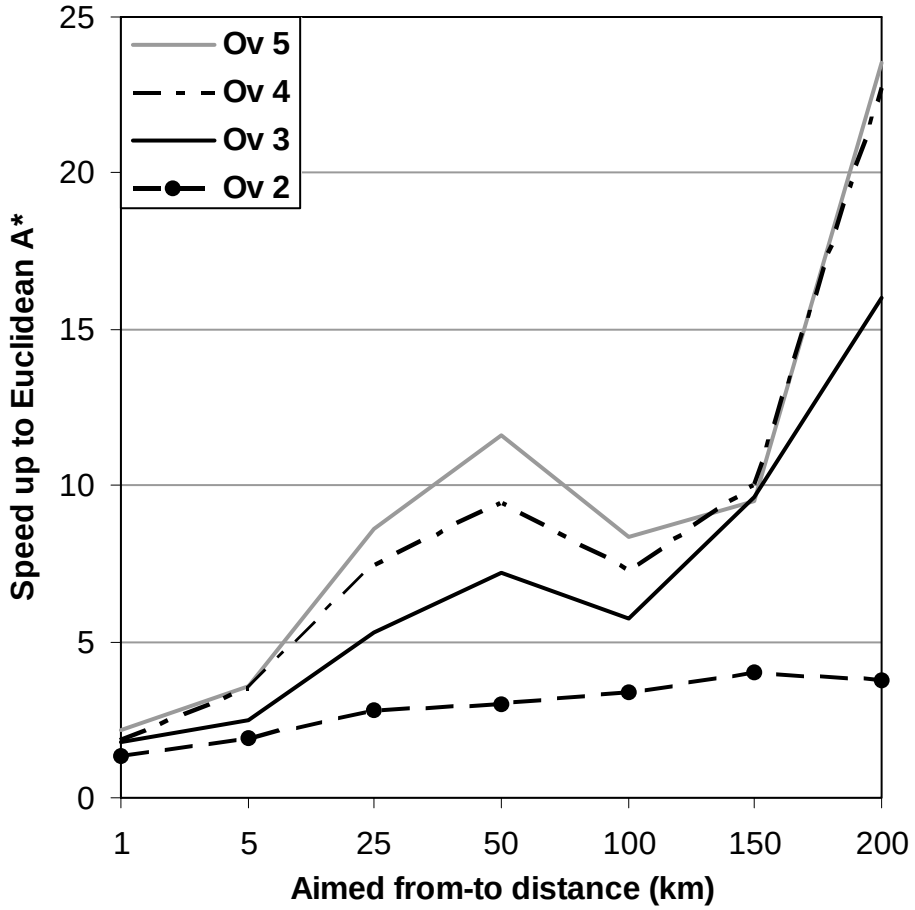
Performance: Loaded (Net 4)

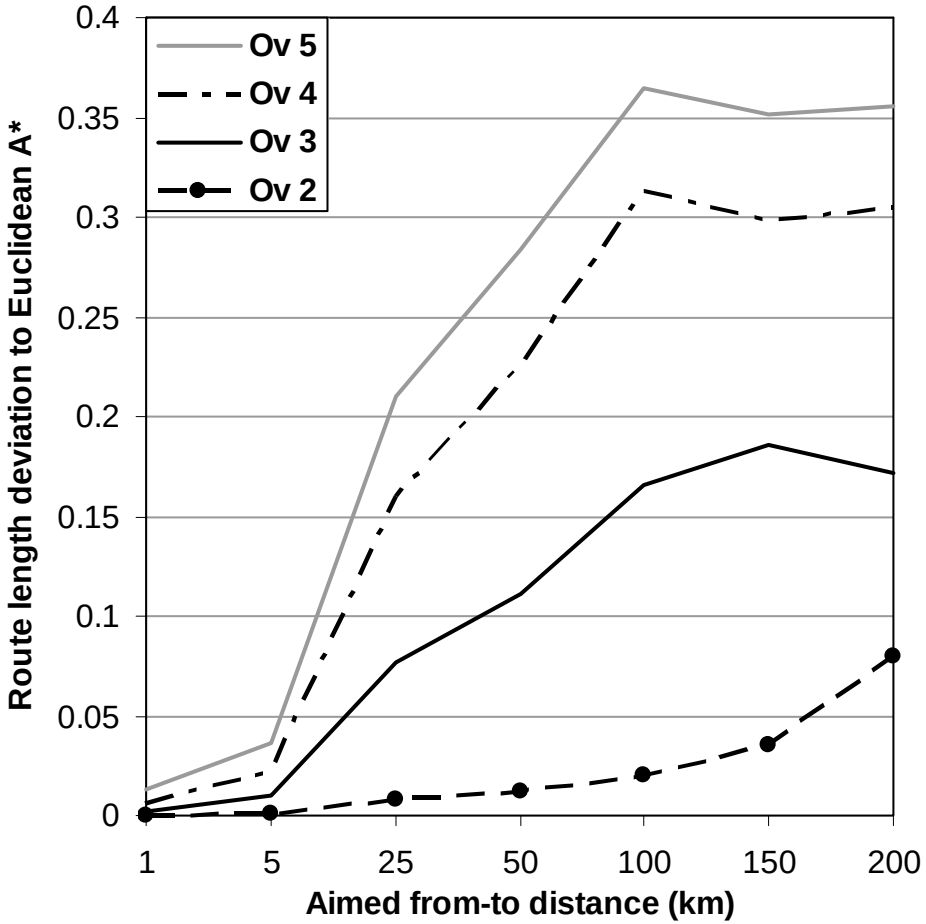


Performance: Landmarks (Free Flow, Net 2)



Performance: Overdo A* Vs. Euc A* (Free Flow, Net 3)

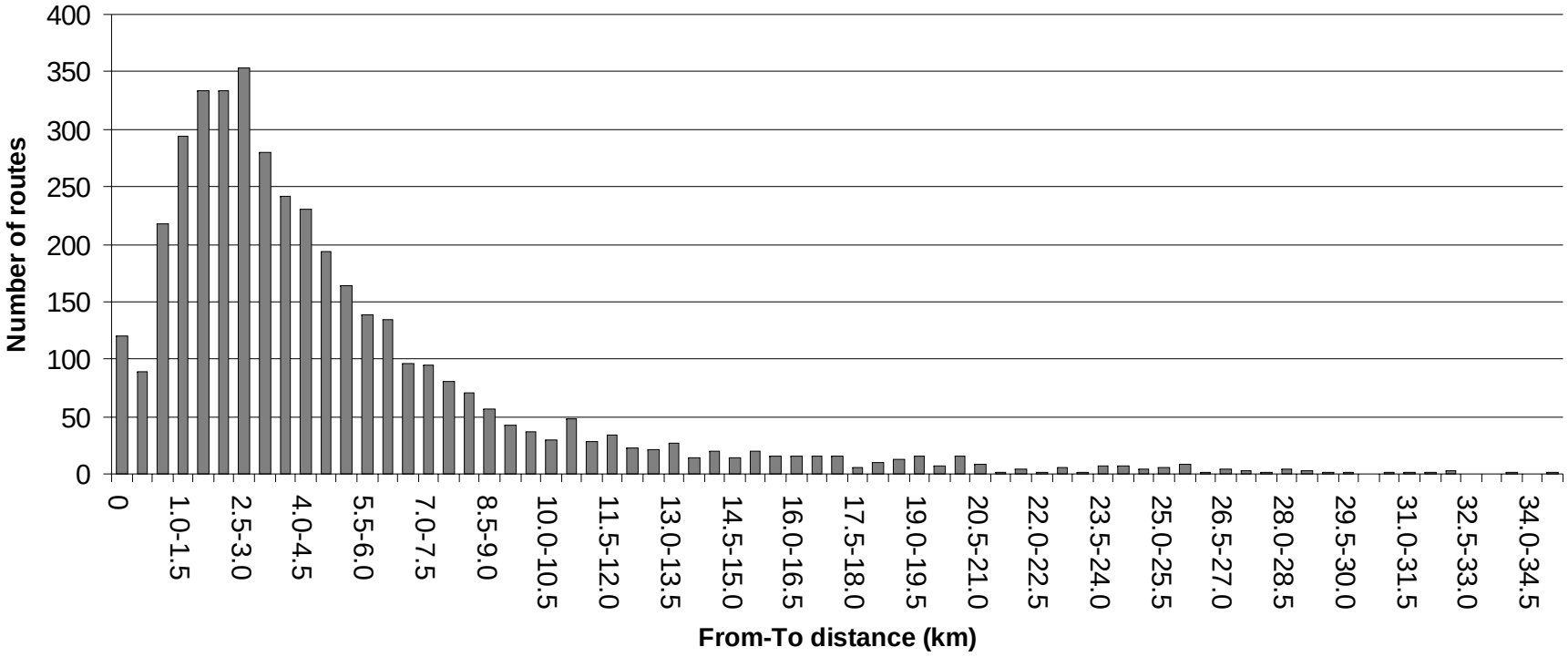




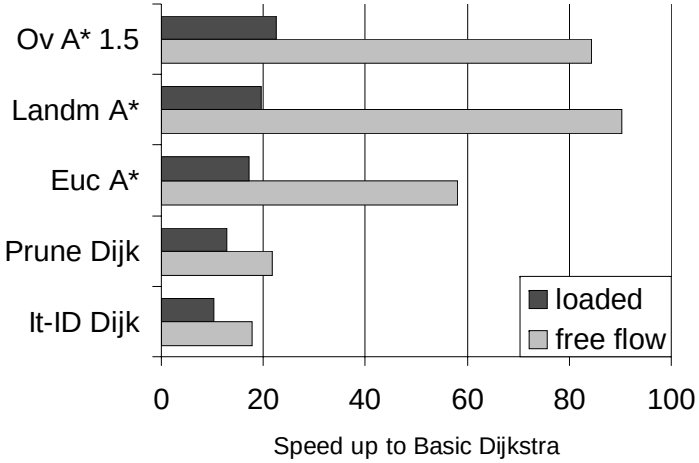
M. Balmer

Performance: Real World Example (1)

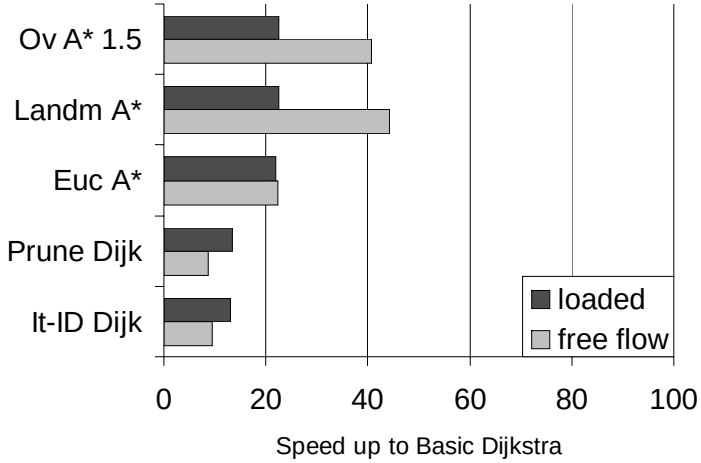
- Greater Zurich area
- Work-commuter population (sample of 10’000 trips)
- Free flow & loaded networks



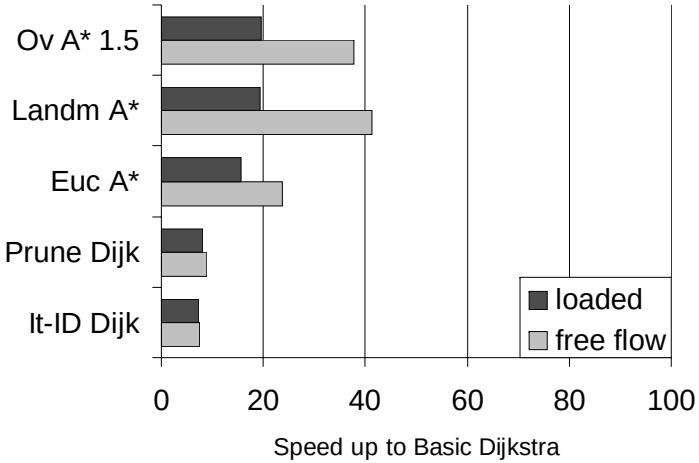
Performance: Real World Example (2)



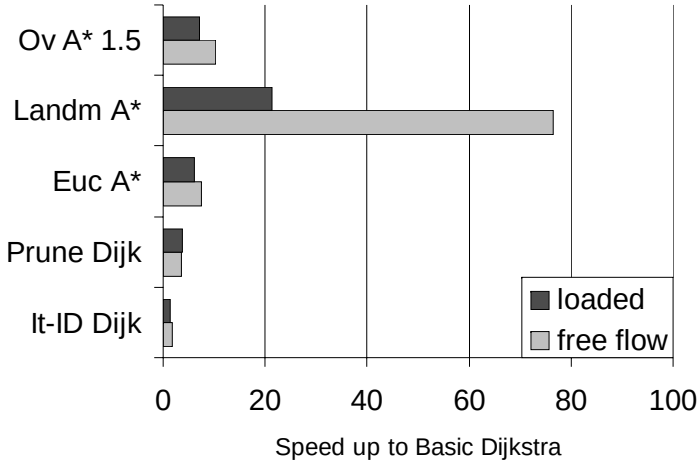
Net 1



Net 2



Net 3



Net 4

Performance: Latest Results

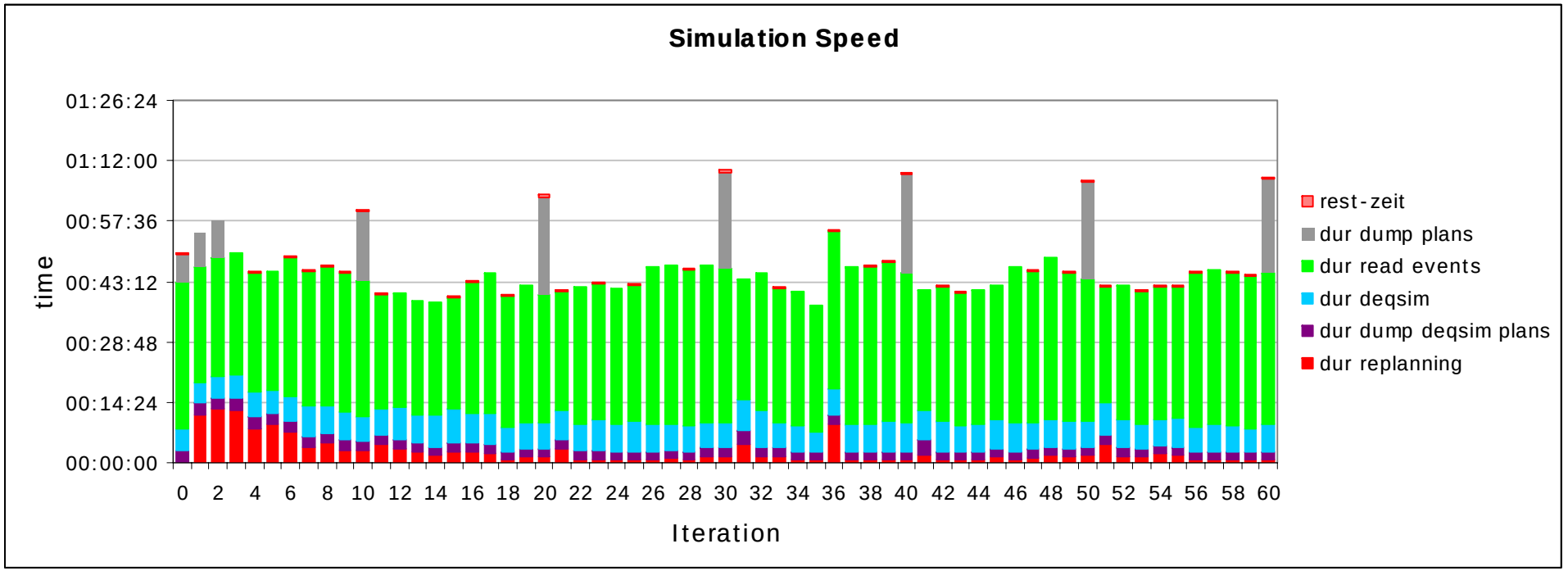
- Whole Switzerland

- Daily demand for ca. 2.3 million persons (ca. 7.6 Mio trips)

- “ivtch” net (national network model, ~60’000 links)

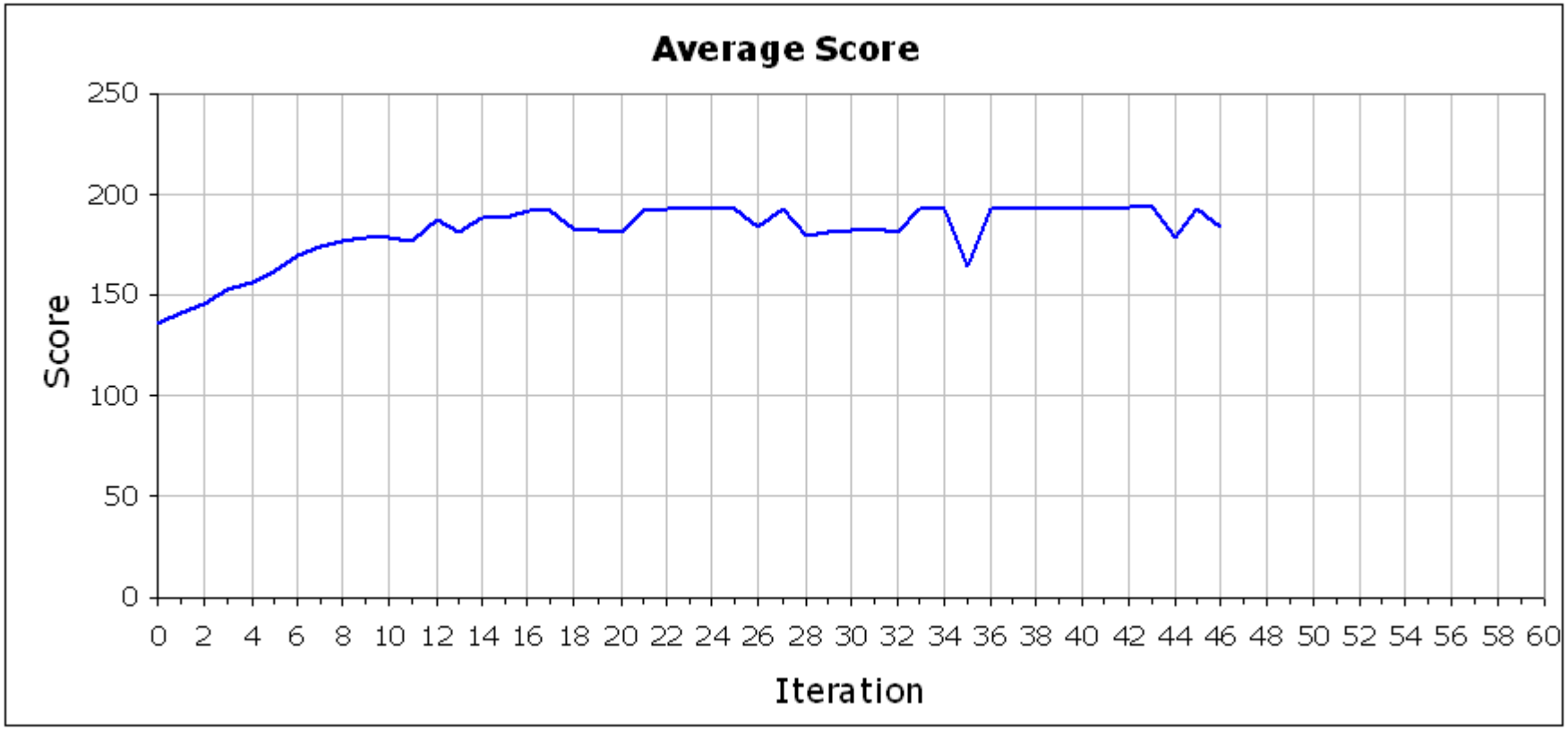
- 46 Iteration processes with MATSim-T (00:00 - 48:00 h)

Performance: Latest Results (2)

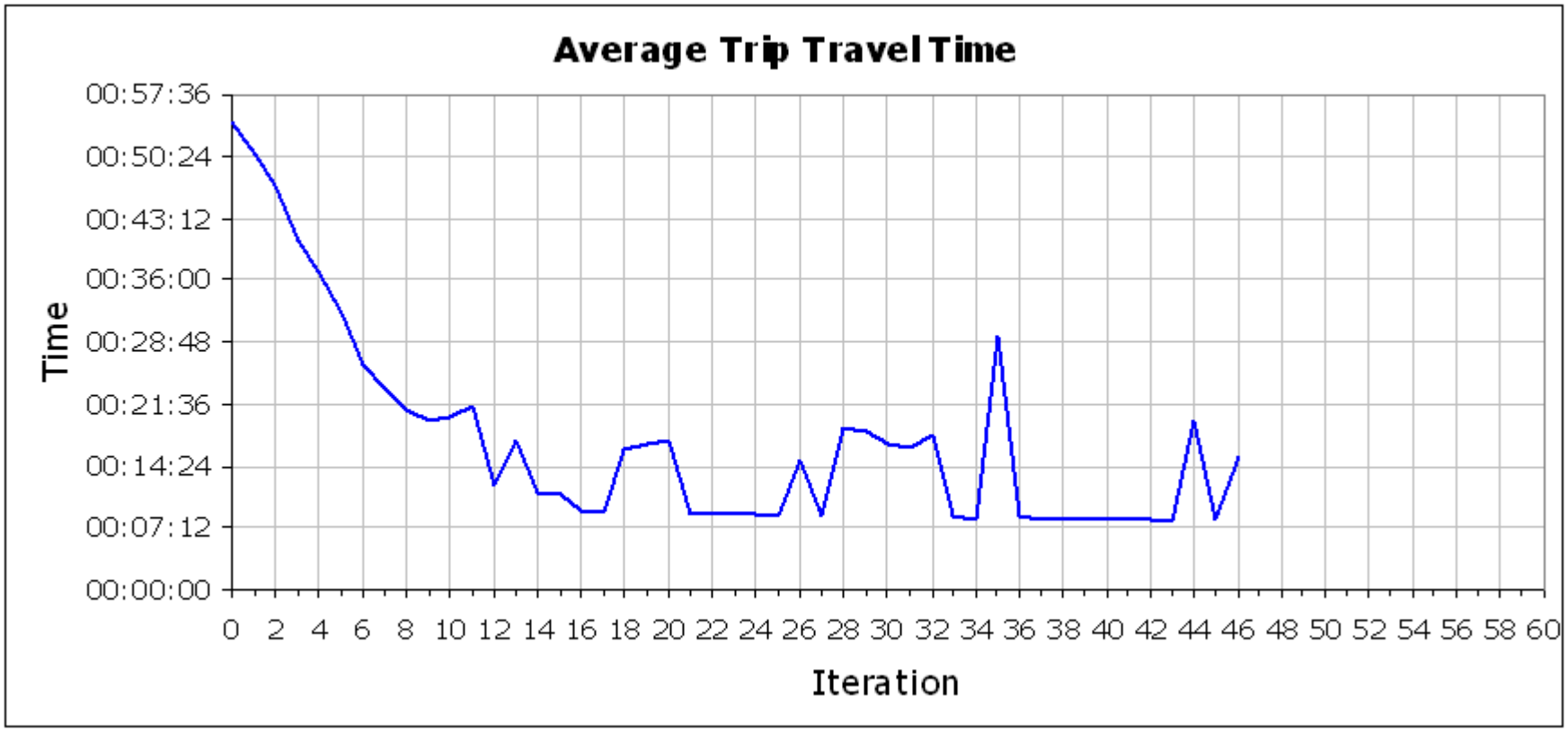


More about the Run...

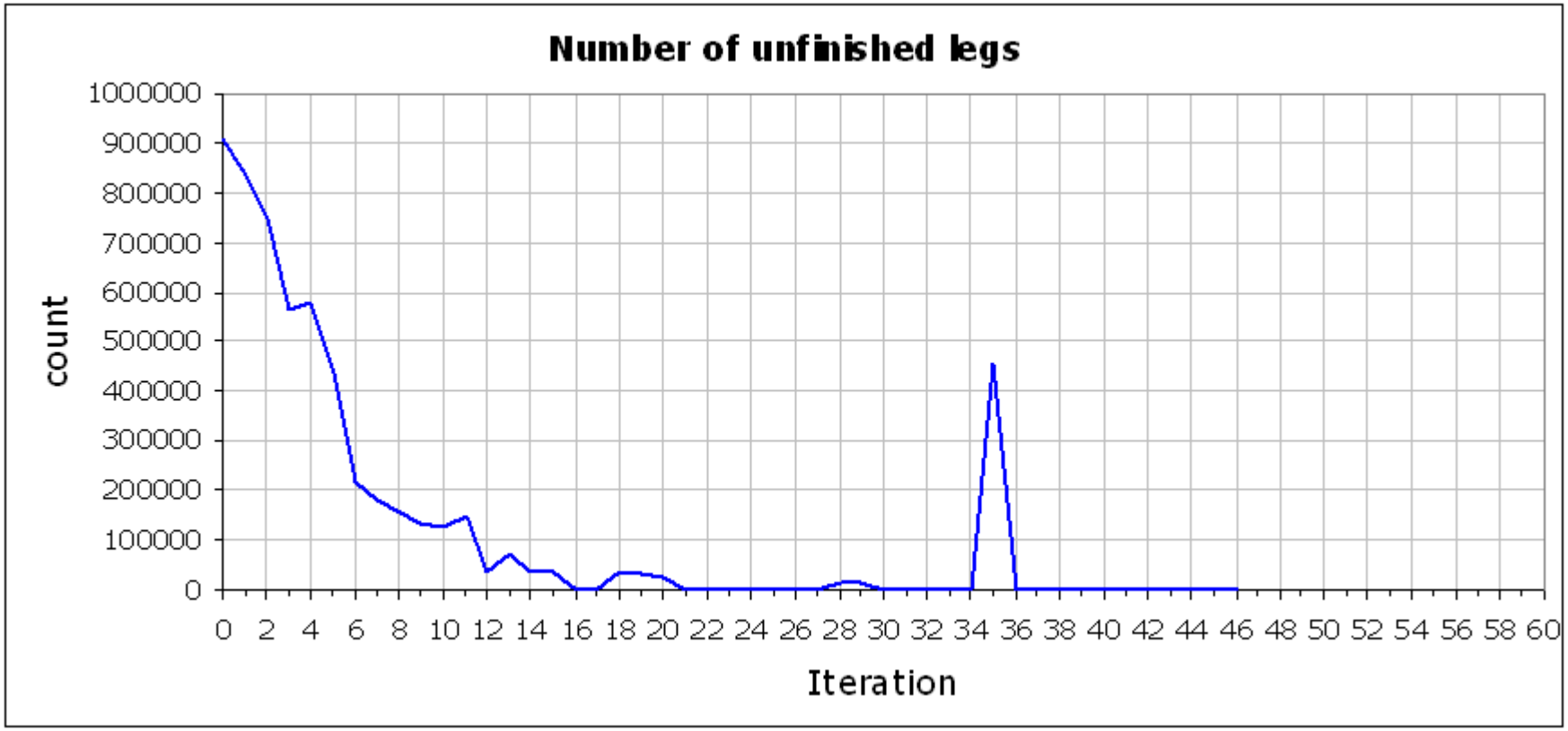
More about the Run...



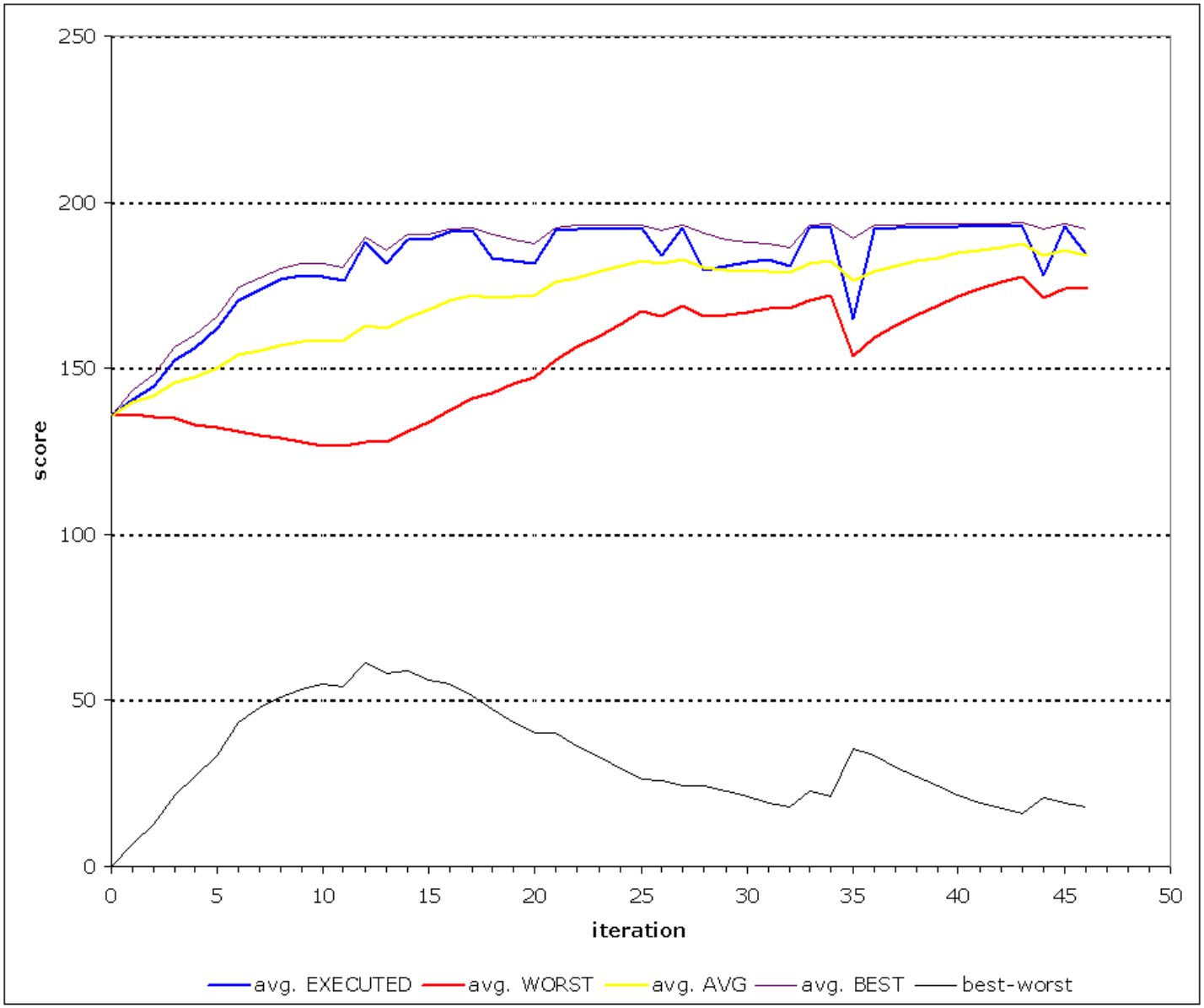
More about the Run...



More about the Run...



More about the Run...



More about the Run...

Departure, arrival, and “on route” histograms (5min time bin)

Thanks!

Literature

Balmer, M. (2007) Travel demand modeling for multi-agent traffic simulations: Algorithms and systems, Ph.D. Thesis, ETH Zurich, Zurich.

Chabini, I. and S. Lan (2002) Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks, *IEEE Transactions on Intelligent Transportation Systems*, **3** (1) 60–74.

Goldberg, A. V. and C. Harrelson (2005) Computing the shortest path: A* search meets graph theory, paper presented at the *16th annual ACM-SIAM symposium on Discrete algorithms (SODA)*, Vancouver, Jan. 2005.

Hart, P. E., N. J. Nilsson and B. Raphael (1972) A formal basis for the heuristic determination of minimum cost paths, *ACM SIGART Bulletin*, **37**, 28–29.

Sedgewick, R. and J. S. Vitter (1986) Shortest paths in Euclidean graphs, *Algorithmica*, **1** (1) 31–48.

Wagner, D. and T. Willhalm (2003) Geometric speed-up techniques for finding shortest paths in large sparse graphs, in G. di Battista and U. Zwick (eds.) *Lecture Notes in Computer Science*, **2832**, 776–787, Springer, Budapest.