

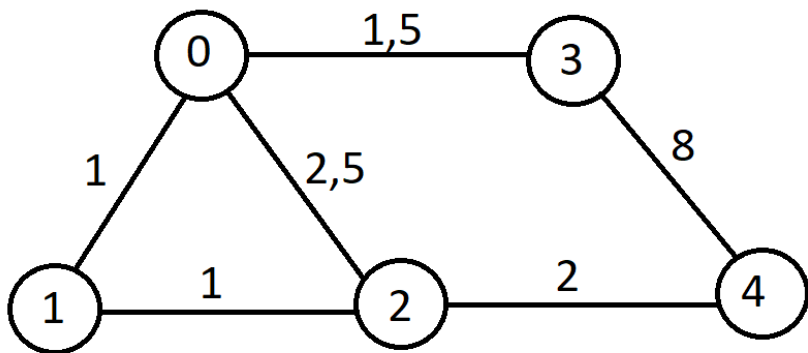
Abschlusspräsentation Projekt 2

Pouria Araghchi 170468, Kai Lukas Ilmenau 225338, Naveed Niazi 214471

June 27, 2023

TU Dortmund - Fachprojekt zu "Routingalgorithmen"

Kai



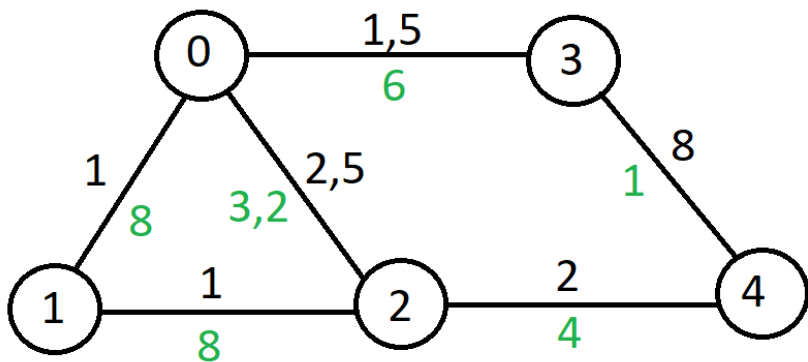
Capacity

InverseCap.

ClosenessInvCap.

ClosenessCentr.

Inverse Capacity



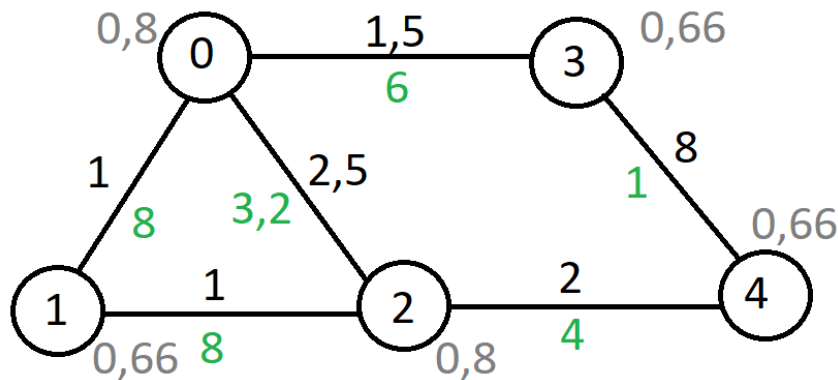
Capacity

InverseCap.

ClosenessInvCap.

ClosenessCentr.

Closeness Centrality Werte



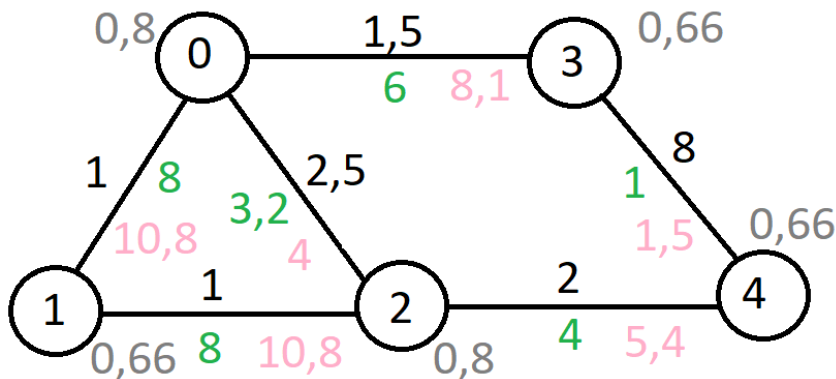
Capacity

InverseCap.

ClosenessInvCap.

ClosenessCentr.

Closeness Centrality With Inverse Capacity



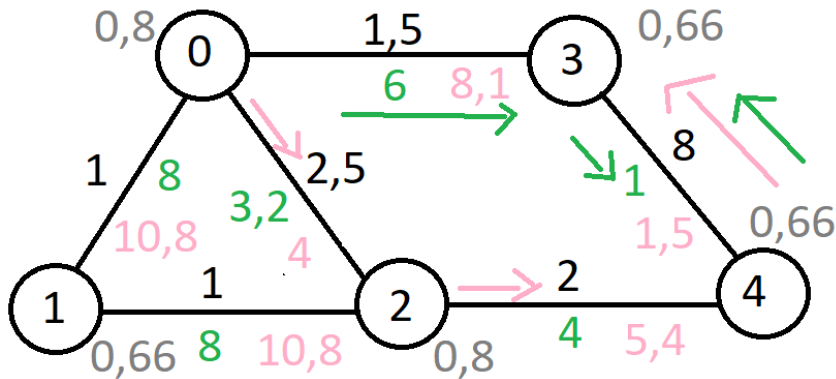
Capacity

InverseCap.

ClosenessInvCap.

ClosenessCentr.

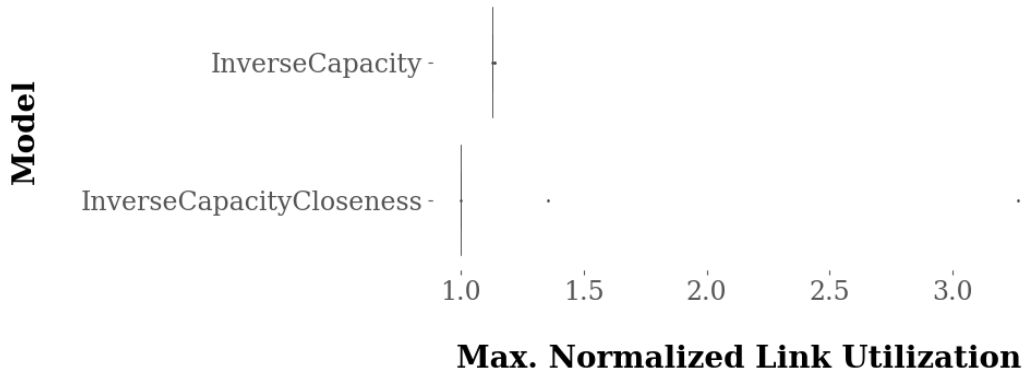
Mit Demands



Capacity
InverseCap.
ClosenessInvCap.
ClosenessCentr.

Demands von 0 nach 4 (eine Capacity-Einheit)
und von 4 nach 3 (acht Capacity-Einheiten)

Boxplots



Ist meine Algorithmus besser?

- in diesem speziellen Setting: Ja
- "designed" Topologie
 - lange Suche nach guter Topologie für den Algorithmus
- Siehe Projekt 1: *Inverse_Capacity* im Durchschnitt besser als meine Anpassung

Naveed

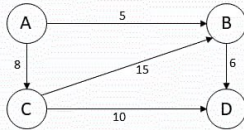
Algorithmus Sequential Combination:

- Nimmt zwei Algorithmen als Parameter OSPF und DFW(Demand First Way Point)
- Ersetzen DFW durch UW(Uniform Weights)

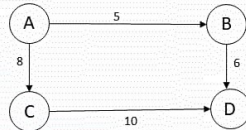
Warum?

- DFW und UW werden zur Optimierung der Verkehrssteuerung in Netzwerk eingesetzt.
- Beide Algorithmen zielen drauf ab die Netzwerküberlastung durch Anpassung der Linkgewichte zu minimieren
- Aber die unterscheiden sich in ihrer Herangehensweise an das Traffic Engineering
- Der DFW-Algorithmus konzentriert sich darauf, den Verkehr zu spezifischen Zwischenpunkten im Netzwerk, sogenannten Waypoints, zu leiten, um die Verkehrslast auszugleichen und Staus zu vermeiden. Und Der UW-Algorithmus hingegen zielt darauf ab, den Verkehr gleichmäßig über das Netzwerk zu verteilen. Die Algorithmen funktionieren wie folgt:

DFW:



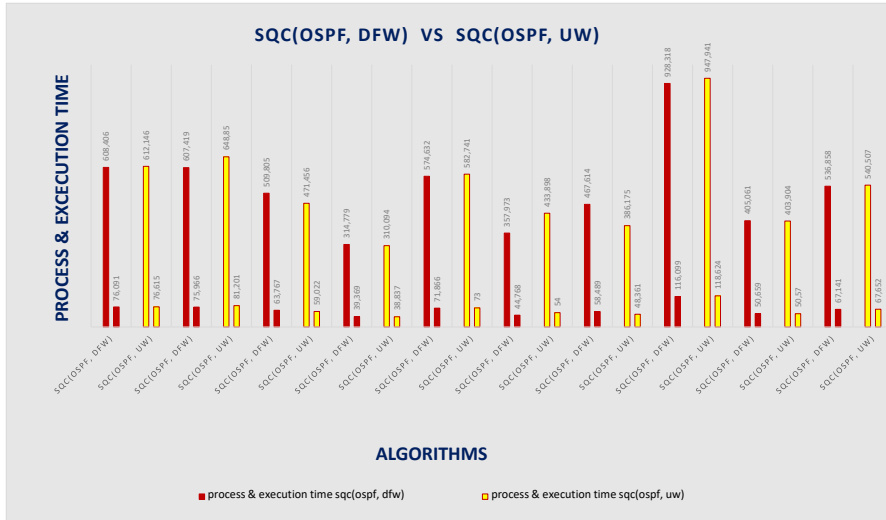
UW:



Ergebnisse

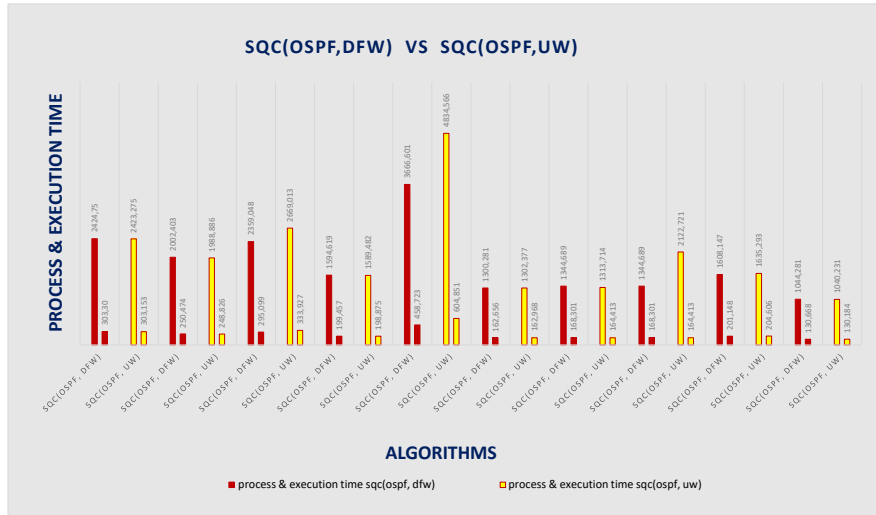
All Algorithms

Algorithm-Name	Topology	Nodes	Links	MCF-Synthetic Damands	Process-Time	Execution-Time
demandFirstWayPoint	abilene	12	30	182	0,509	0,064
uniformWeights	abilene	12	30	182	0,074	0,074
heur_ospf_weights	abilene	12	30	182	601,031	75,168
sq_demandFirstWayPoint	abilene	12	30	182	608,406	76,091
sq_uniformWeight	abilene	12	30	182	612,146	76,615
demandFirstWayPoint	abilene	12	30	182	0,465	0,058
uniformWeights	abilene	12	30	182	0,057	0,057
heur_ospf_weights	abilene	12	30	182	612,31	76,579
sq_demandFirstWayPoint	abilene	12	30	182	607,419	75,966
sq_uniformWeight	abilene	12	30	182	648,85	81,201
demandFirstWayPoint	abilene	12	30	182	0,47	0,059
uniformWeights	abilene	12	30	182	0,069	0,069
heur_ospf_weights	abilene	12	30	182	572,566	71,619
sq_demandFirstWayPoint	abilene	12	30	182	509,805	63,767
sq_uniformWeight	abilene	12	30	182	471,456	59,022
demandFirstWayPoint	abilene	12	30	182	0,496	0,062
uniformWeights	abilene	12	30	182	0,636	0,064
heur_ospf_weights	abilene	12	30	182	309,209	39
sq_demandFirstWayPoint	abilene	12	30	182	314,779	39,369
sq_uniformWeight	abilene	12	30	182	310,094	38,837
demandFirstWayPoint	abilene	12	30	182	0,543	0,068
uniformWeights	abilene	12	30	182	0,0605	0,061
heur_ospf_weights	abilene	12	30	182	582,1	72,801
sq_demandFirstWayPoint	abilene	12	30	182	574,632	71,866
sq_uniformWeight	abilene	12	30	182	582,741	73
demandFirstWayPoint	abilene	12	30	182	0,474	0,059
uniformWeights	abilene	12	30	182	0,074	0,074
heur_ospf_weights	abilene	12	30	182	356,241	44,551
sq_demandFirstWayPoint	abilene	12	30	182	357,973	44,768
sq_uniformWeight	abilene	12	30	182	433,898	54
demandFirstWayPoint	abilene	12	30	182	0,735	0,092
uniformWeights	abilene	12	30	182	0,087	0,087
heur_ospf_weights	abilene	12	30	182	513,129	64,183
sq_demandFirstWayPoint	abilene	12	30	182	467,614	58,489
sq_uniformWeight	abilene	12	30	182	386,175	48,361
demandFirstWayPoint	abilene	12	30	182	0,484	0,061
uniformWeights	abilene	12	30	182	0,061	0,061
heur_ospf_weights	abilene	12	30	182	926,012	115,811
sq_demandFirstWayPoint	abilene	12	30	182	928,318	116,099
sq_uniformWeight	abilene	12	30	182	947,941	118,624
demandFirstWayPoint	abilene	12	30	182	0,467	0,058
uniformWeights	abilene	12	30	182	0,073	0,073
heur_ospf_weights	abilene	12	30	182	414,148	51,796
sq_demandFirstWayPoint	abilene	12	30	182	405,061	50,659
sq_uniformWeight	abilene	12	30	182	403,904	50,57
demandFirstWayPoint	abilene	12	30	182	0,465	0,058
uniformWeights	abilene	12	30	182	0,058	0,058
heur_ospf_weights	abilene	12	30	182	539,813	67,512
sq_demandFirstWayPoint	abilene	12	30	182	536,858	67,141
sq_uniformWeight	abilene	12	30	182	540,507	67,652



Results-Real-Demands

Algorithm-Name	Topology	Nodes	Links	MCF-Synthetic-Demands	Process-Time	Execution-Time
demandFirstWayPoint	abilene	12	30	917	1,505	0,227
uniformWeights	abilene	12	30	917	0,006	0,006
heur_ospf_weights	abilene	12	30	917	2406,44	300,93
sq_demandFirstWayPoint	abilene	12	30	917	2424,75	303,30
sq_uniformWeight	abilene	12	30	917	2423,275	303,153
demandFirstWayPoint	abilene	12	30	924	1,505	0,226
uniformWeights	abilene	12	30	924	0,006	0,006
heur_ospf_weights	abilene	12	30	924	1988,597	248,701
sq_demandFirstWayPoint	abilene	12	30	924	2002,403	250,474
sq_uniformWeight	abilene	12	30	924	1988,886	248,826
demandFirstWayPoint	abilene	12	30	910	1,539	0,261
uniformWeights	abilene	12	30	910	0,006	0,006
heur_ospf_weights	abilene	12	30	910	2242,128	280,409
sq_demandFirstWayPoint	abilene	12	30	910	2359,048	295,099
sq_uniformWeight	abilene	12	30	910	2669,013	333,927
demandFirstWayPoint	abilene	12	30	763	1,499	0,219
uniformWeights	abilene	12	30	763	0,074	0,074
heur_ospf_weights	abilene	12	30	763	2362,522	295,505
sq_demandFirstWayPoint	abilene	12	30	763	1594,619	199,457
sq_uniformWeight	abilene	12	30	763	1589,482	198,875
demandFirstWayPoint	abilene	12	30	917	1,571	0,247
uniformWeights	abilene	12	30	917	0,006	0,006
heur_ospf_weights	abilene	12	30	917	3086,212	386,005
sq_demandFirstWayPoint	abilene	12	30	917	3666,601	458,723
sq_uniformWeight	abilene	12	30	917	4834,566	604,851
demandFirstWayPoint	abilene	12	30	784	1,539	0,221
uniformWeights	abilene	12	30	784	0,006	0,006
heur_ospf_weights	abilene	12	30	784	1331,053	166,468
sq_demandFirstWayPoint	abilene	12	30	784	1300,281	162,656
sq_uniformWeight	abilene	12	30	784	1302,377	162,968
demandFirstWayPoint	abilene	12	30	924	1,519	0,238
uniformWeights	abilene	12	30	924	0,006	0,006
heur_ospf_weights	abilene	12	30	924	1171,284	146,487
sq_demandFirstWayPoint	abilene	12	30	924	1344,689	168,301
sq_uniformWeight	abilene	12	30	924	1313,714	164,413
demandFirstWayPoint	abilene	12	30	924	1,548	0,269
uniformWeights	abilene	12	30	924	0,006	0,006
heur_ospf_weights	abilene	12	30	924	2365,246	295,823
sq_demandFirstWayPoint	abilene	12	30	924	1344,689	168,301
sq_uniformWeight	abilene	12	30	924	2122,721	164,413
demandFirstWayPoint	abilene	12	30	714	1,479	0,198
uniformWeights	abilene	12	30	714	0,006	0,006
heur_ospf_weights	abilene	12	30	714	1546,135	193,369
sq_demandFirstWayPoint	abilene	12	30	714	1608,147	201,148
sq_uniformWeight	abilene	12	30	714	1635,293	204,606
demandFirstWayPoint	abilene	12	30	924	1,517	0,237
uniformWeights	abilene	12	30	924	0,006	0,006
heur_ospf_weights	abilene	12	30	924	1042,781	130,416
sq_demandFirstWayPoint	abilene	12	30	924	1044,281	130,668
sq_uniformWeight	abilene	12	30	924	1040,231	130,184



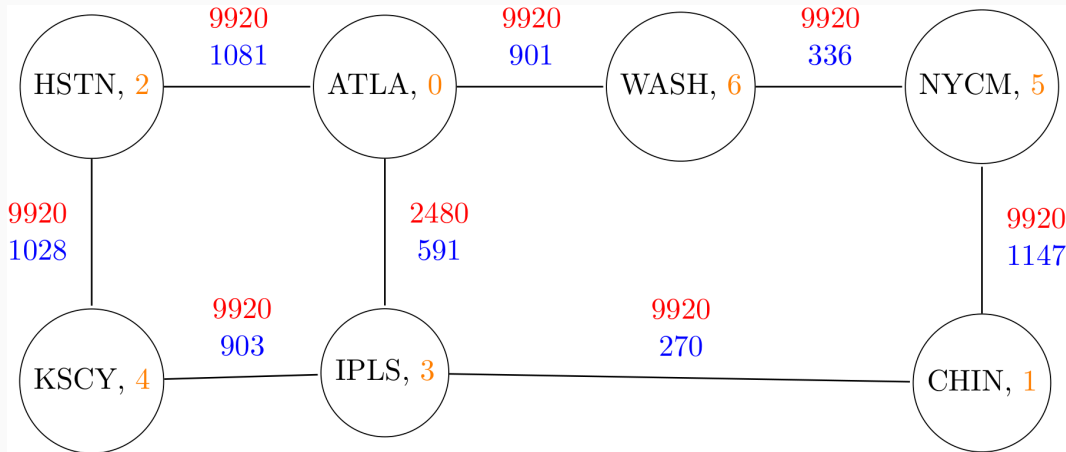
OSPF mit Uniform Weight:

- Einfache Konfiguration und Verwaltung des Routers
- Gleichmäßige Lastverteilung über das Netzwerk wegen der gleichen Kostenmetrik
- Begrenzte Optimierungsfähigkeit: Da keine differenzierten Gewichte verwendet werden, kann OSPF mit uniformen Gewichten keine spezifischen Leistungsmerkmale oder Netzwerkanforderungen berücksichtigen. Dies kann zu Suboptimalität in Bezug auf Bandbreite, Verzögerung oder anderen Faktoren führen, die in bestimmten Anwendungsfällen wichtig sind.
- In komplexen Netzwerken mit unterschiedlichen Leistungseigenschaften der Verbindungen nicht empfehlenswert
- Empfohlen in Netzwerken mit homogenen Verbindungen, bei denen keine signifikante Unterschiede in den Leistungseigenschaften bestehen

Wenn eine dynamische Anpassung des Routings, Ressourceneffizienz und die Fähigkeit, auf Veränderungen zu reagieren, wichtig sind, kann OSPF mit DFWP geeignet sein. Wenn hingegen eine einfache Konfiguration und gleichmäßige Lastverteilung gewünscht sind, kann OSPF mit UW angemessen sein.

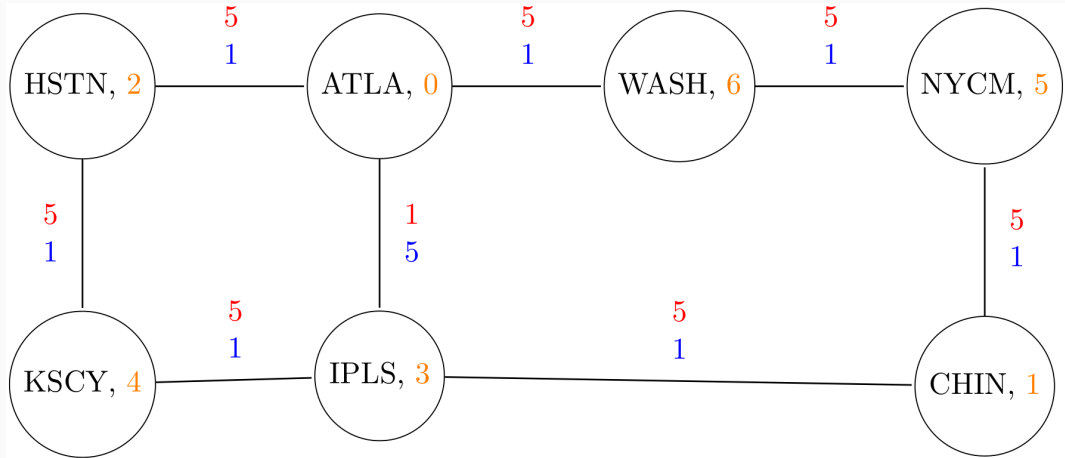
Sequential Combination aus InverseCapacity und DemandFirstWaypoints

Netzwerktopologie (Anlehnung an Abilene)¹



¹FaPro_P2/pouria/network_origin.pdf

Anwendung von InverseCapacity²



²FaPro_P2/pouria/network_origin.pdf

Demandtabelle (vereinfacht)³

↓ von, nach →	HSTN	KSCY	ATLA	IPLS	CHIN	WASH	NYCM
HSTN		1	13	2	6	8	3
KSCY	1		1	2	4	2	2
ATLA	28	1		5	3	19	4
IPLS	7	2	4		14	7	9
CHIN	165	17	18	7		11	12
WASH	13	5	13	5	17		20
NYCM	16	3	9	4	61	24	

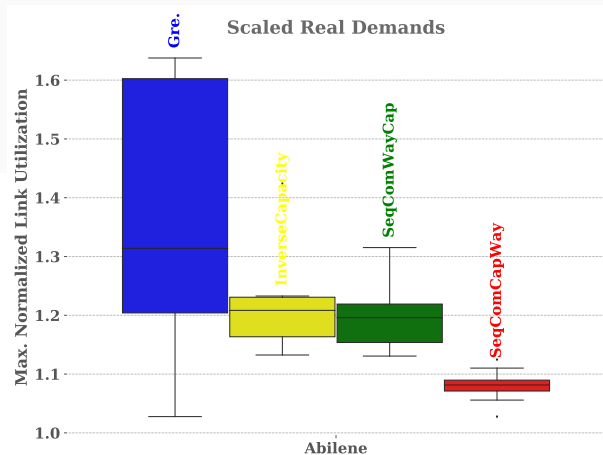
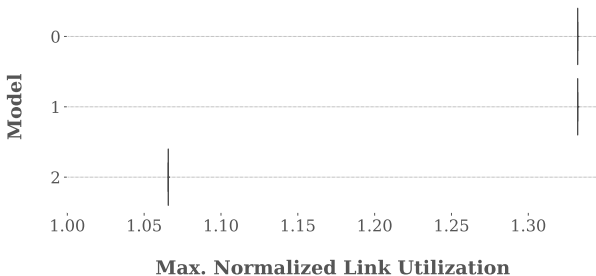
³FaPro_P2/pouria/network_origin.pdf

Demandtabelle (stärker vereinfacht)⁴

↓ von, nach →	HSTN	KSCY	ATLA	IPLS	CHIN	WASH	NYCM
HSTN							
KSCY							
ATLA				5			
IPLS						7	
CHIN							
WASH							
NYCM							

⁴FaPro_P2/pouria/network_origin.pdf

Resultierender Boxplot



- keine Boxen:
 - vermutlich zu schwacher Rechner (2 Minuten Timeout reicht nicht)
- Sequential Combination besser:
 - meine Topologie wurde vereinfacht, von Grund auf (**.json**→**.topo.py**→**.topo.sh**)
 - Joint/Weights evtl nicht

Ergebnisse Reproduktion von Gruppe 3

main: process killed itself

daniel: Process killed itself

daniel: Dependency nicht in README

Fragen oder Anmerkungen?