technische universität
dortmund

Abschlusspräsentation Projekt 1

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

May 23, 2023

TU Dortmund - Fachprojekt zu ”Routingalgorithmen”

Inverse Capacity mit Zentralitätsmerkmalen

Inverse Capacity mit Zentralitätsmerkmalen

- inverseCapacity weights werden mit den Zentralitäten der Knoten einer Kante verrechnet
    - $weight * \frac{CentralityNode_i + CentralityNode_j}{2}$
- untersuchte Zentralitätensmetriken:
    - Betweenness, Closeness, Eigenvevtor

Welche Zentralitätsmetriken?

Closeness centrality

- wie nah ein Knoten zu den anderen ist
- $\mathcal{P}_{i\to j}$ ist der kürzeste Pfad von i nach j
- $H(\mathcal{P}_{i\to j})$ ist der Hop-Count des Pfades

$$c_i = \frac{1}{\sum_{j\neq i} H(\mathcal{P}_{i\to j})}$$

Betweenness centrality

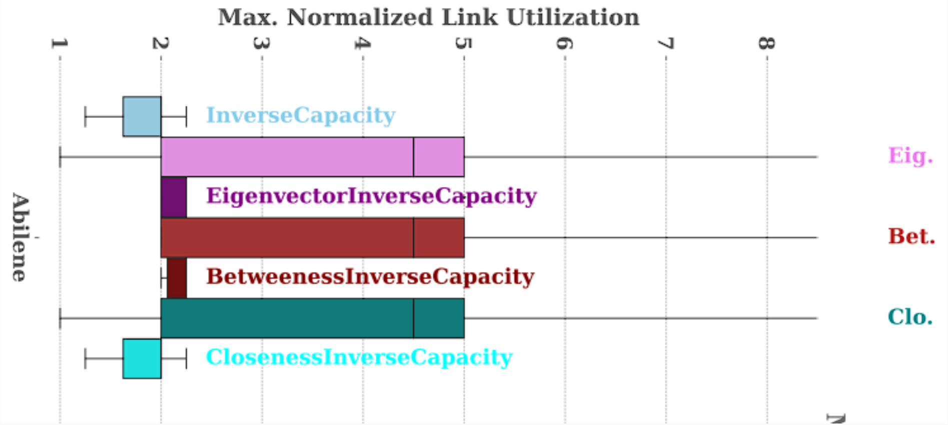
- Verhältnis aller kürzeren Wege zur Anzahl der kürzesten Wege die durch den Knoten gehen

$$b_i = \sum_{s,t\in\mathcal{N}} \frac{|\mathcal{P}_{i\to j}(i)|}{|\mathcal{P}_{i\to j}|}$$

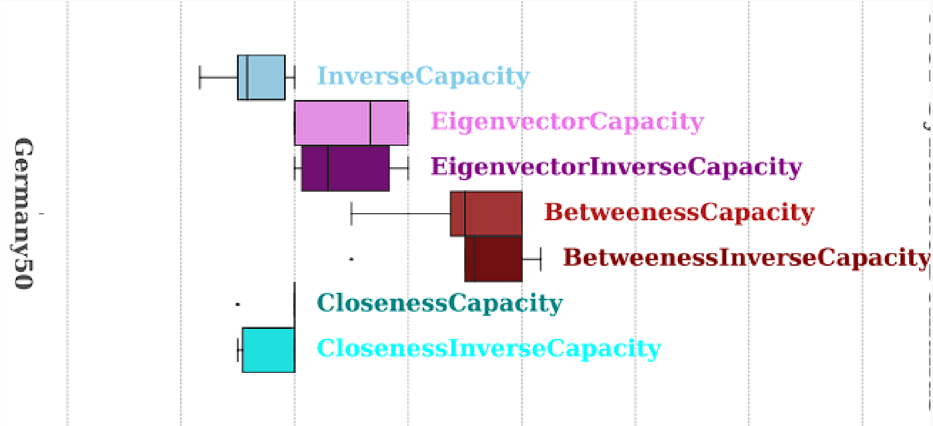
Eigenvector centrality

- entspricht dem i-ten Element des Eigenvektors der dem größten Eigenwert λ_1 der Adjazenzmatrix entspricht

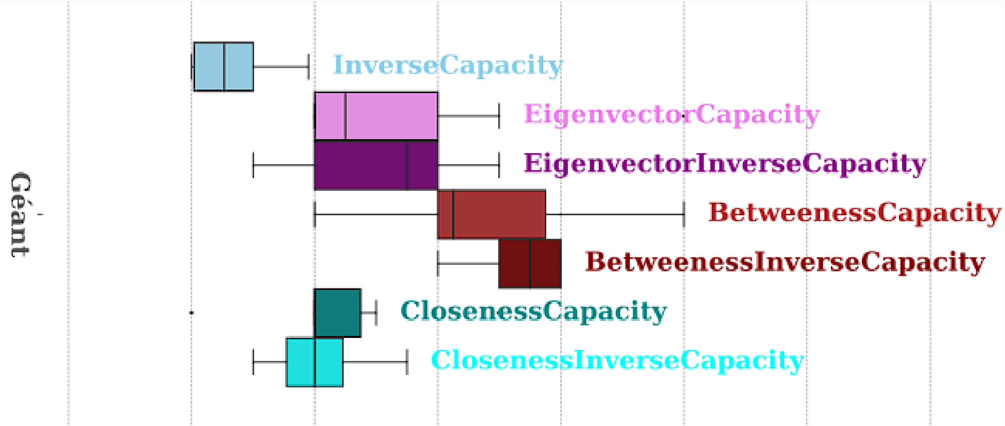
Ergebnisse


Max. Normalized Link Utilization
1
2
3
4
5
6
7
8
Abilene
InverseCapacity
Eig.
EigenvectorInverseCapacity
Bet.
BetweenessInverseCapacity
Clo.
ClosenessInverseCapacity

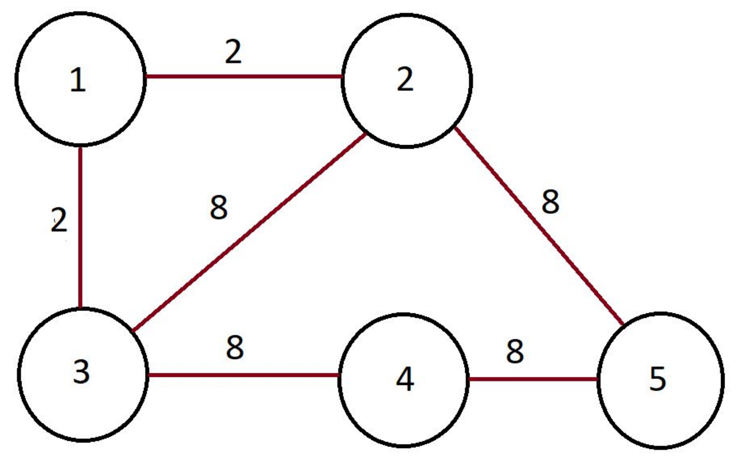
Ergebnisse


Germany50
InverseCapacity
EigenvectorCapacity
EigenvectorInverseCapacity
BetweenessCapacity
BetweenessInverseCapacity
ClosenessCapacity
ClosenessInverseCapacity

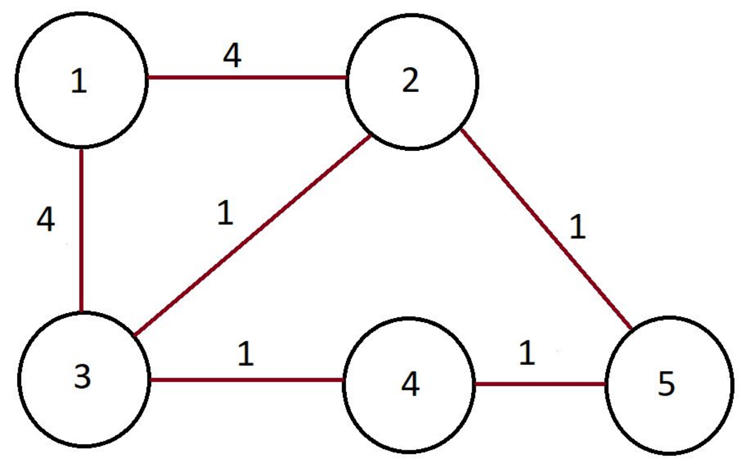
Ergebnisse


Géant
InverseCapacity
EigenvectorCapacity
EigenvectorInverseCapacity
BetweenessCapacity
BetweenessInverseCapacity
ClosenessCapacity
ClosenessInverseCapacity

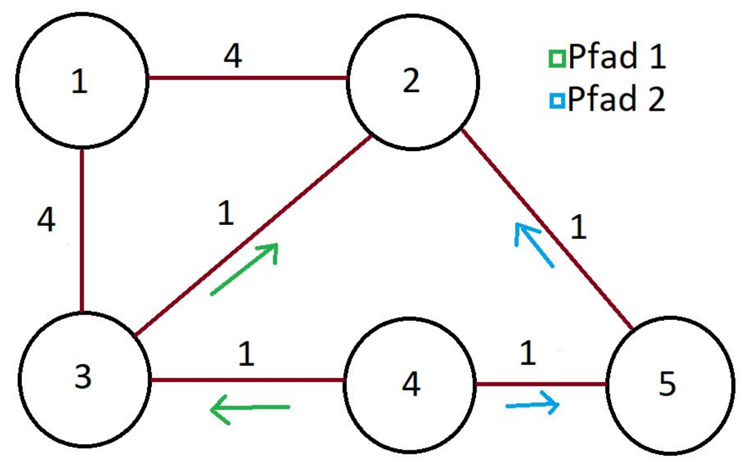
Erklärung



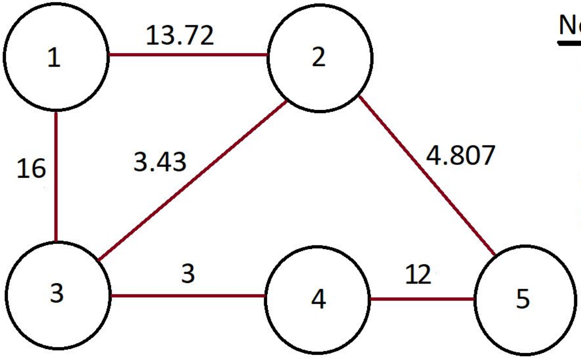
InverseCapacity anwenden



Optimale Pfade

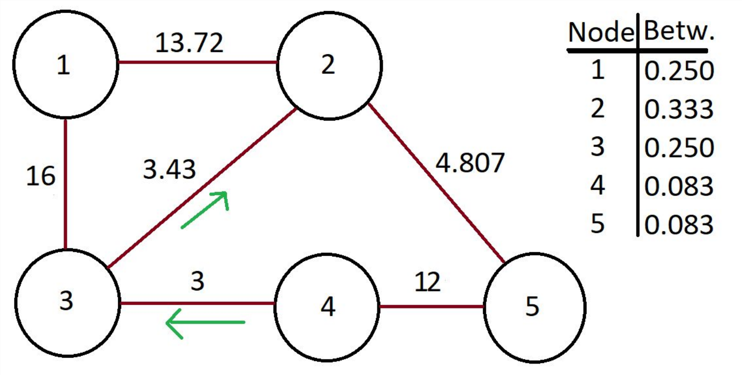


Jetzt mit Zentralitäten



Node	Betw.
1	0.250
2	0.333
3	0.250
4	0.083
5	0.083

Anzahl der Pfade

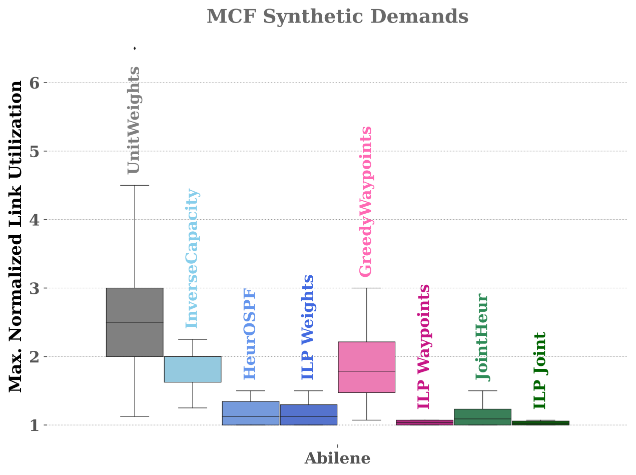


Node	Betw.
1	0.250
2	0.333
3	0.250
4	0.083
5	0.083

Naveed

Plotergebnisse

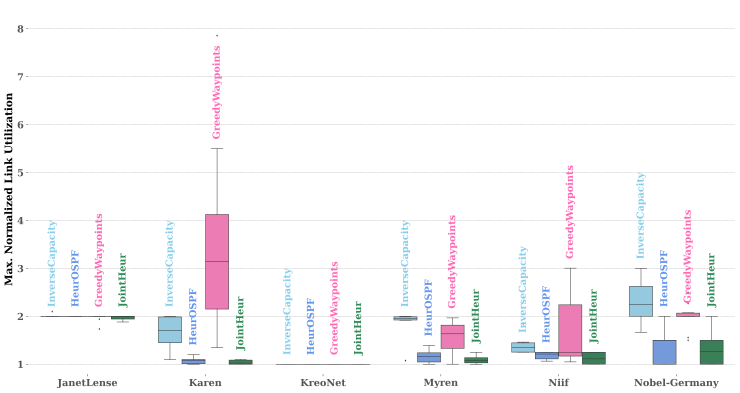
- All-algorithms



Plotergebnisse

- All-topologies:

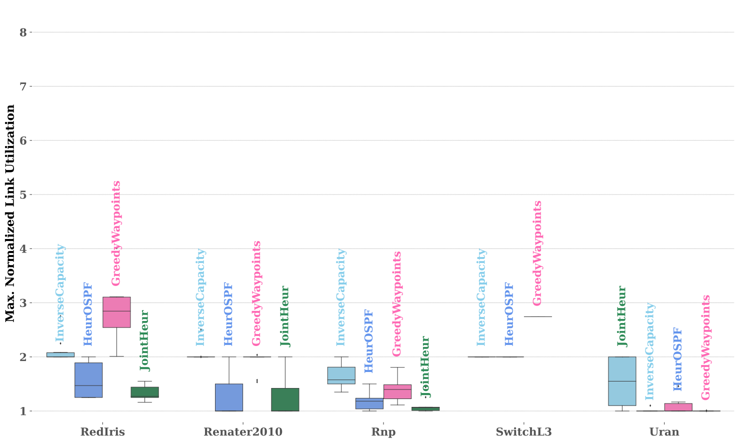
  1)



Plotergebnisse

- All-topologies:

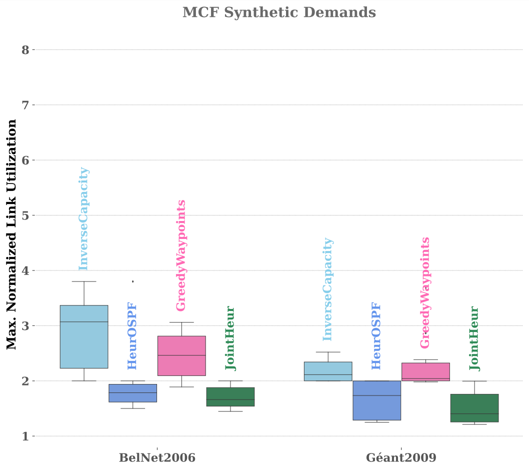
  2)



Plotergebnisse

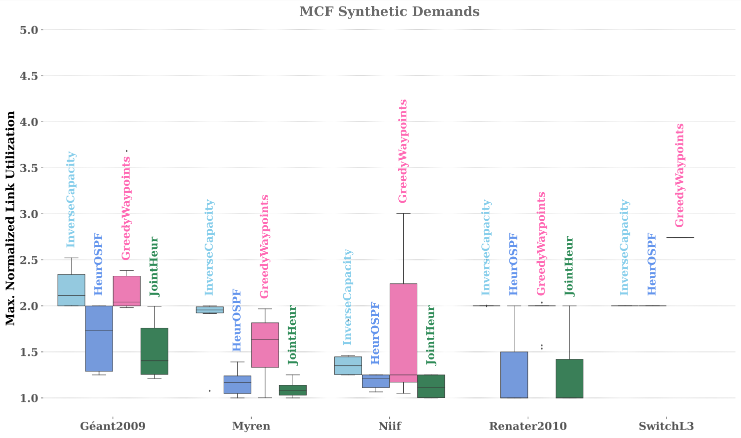
- All-topologies:

  3)



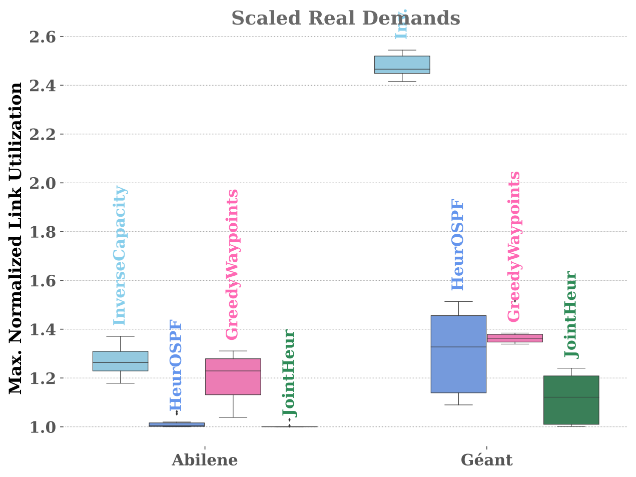
Plotergebnisse

- All-topologies:
  4)

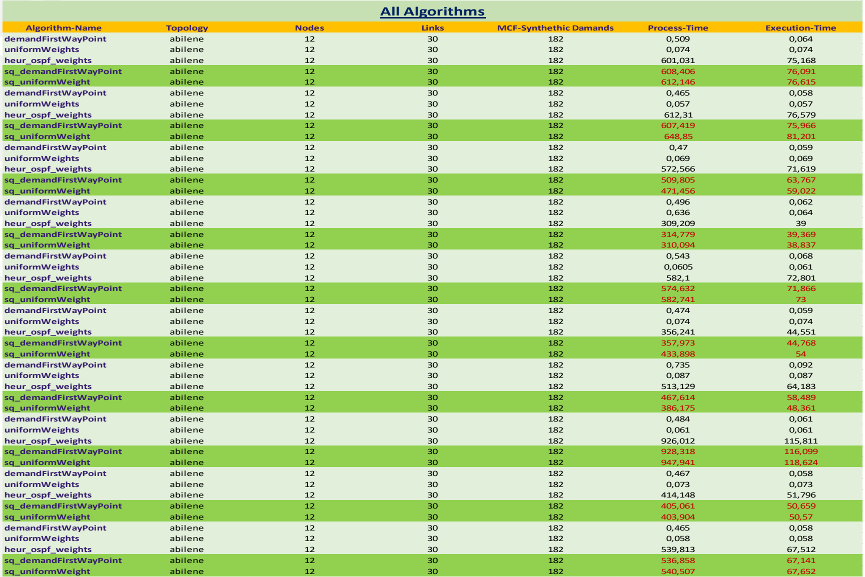


Plotergebnisse

- Real demands:



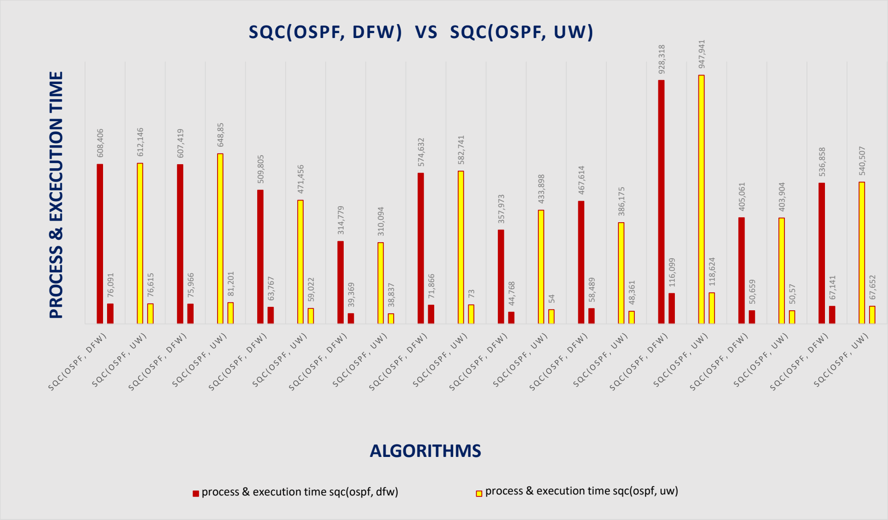
Ergebnisse



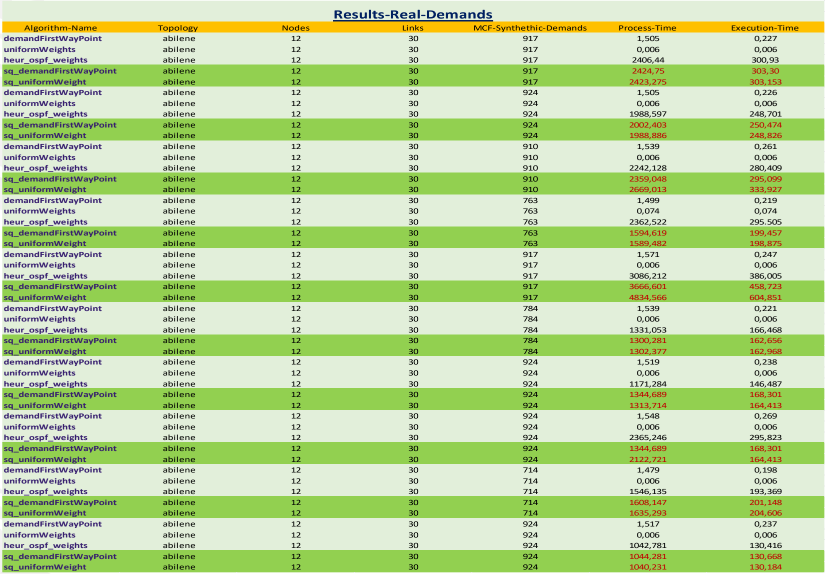
All Algorithms

Algorithm-Name	Topology	Nodes	Links	MCF-Synthethic 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

Ergebnisse


SQC(OSPF, DFW) VS SQC(OSPF, UW)
PROCESS & EXCECUTION TIME
608,406
76,091
612,146
76,615
607,419
75,966
648,85
81,201
509,805
63,767
471,456
59,022
314,779
39,369
310,094
38,837
574,632
71,866
582,741
73
357,973
44,768
433,898
54
467,614
58,489
386,175
48,361
928,318
116,099
947,941
118,624
405,061
50,659
403,904
50,57
536,858
67,141
540,507
67,652
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
ALGORITHMS
🟥 process & execution time sqc(ospf, dfw)
🟨 process & execution time sqc(ospf, uw)

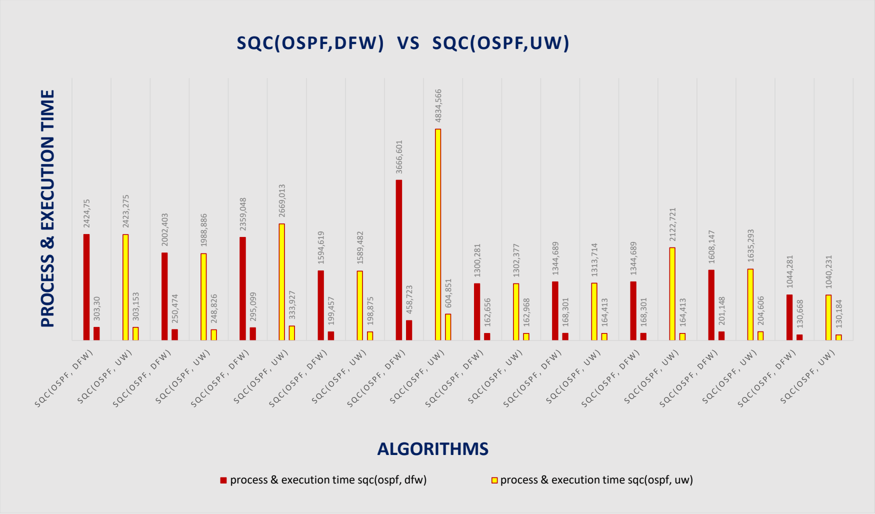
Ergebnisse



Results-Real-Demands

Algorithm-Name	Topology	Nodes	Links	MCF-Synthethic-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

Ergebnisse


SQC(OSPF,DFW) VS SQC(OSPF,UW)
PROCESS & EXECUTION TIME
2424,75
303,30
2423,275
303,153
2002,403
250,474
1988,886
248,826
2359,048
295,099
2669,013
333,927
1594,619
199,457
1589,482
198,875
3666,601
458,723
4834,566
604,851
1300,281
162,656
1302,377
162,968
1344,689
168,301
1313,714
164,413
1344,689
168,301
2122,721
164,413
1608,147
201,148
1635,293
204,606
1044,281
130,668
1040,231
130,184
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
SQC(OSPF, DFW)
SQC(OSPF, UW)
ALGORITHMS
🟥 process & execution time sqc(ospf, dfw)
🟨 process & execution time sqc(ospf, uw)

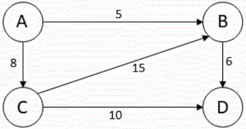
Algorithmus Sequential_Combination:

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

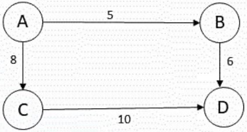
Warum?

- ➢ 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:



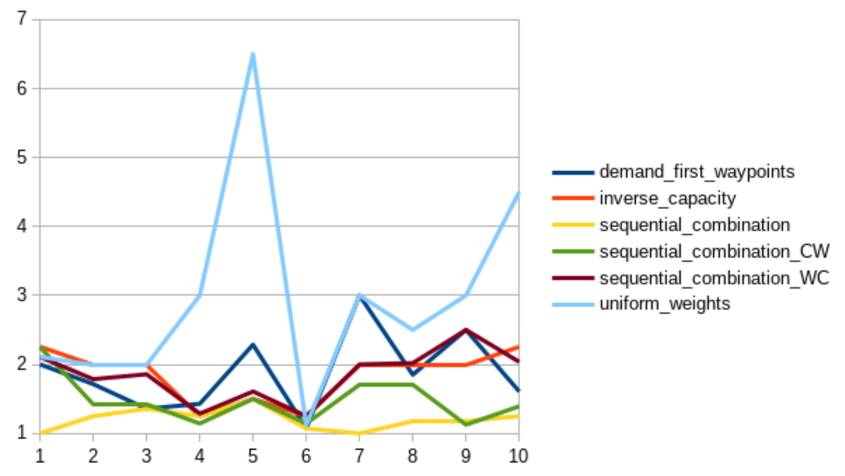
Sequential Combination aus InverseCapacity und DemandFirstWaypoints

Zweiter Schritt: Motivation

- wenig Rechenzeit für *inverse_capacity* und *demand_first_waypoints*
- Kombi aus beidem genauer?
- zusätzliche Rechenzeit gerechtfertigt?

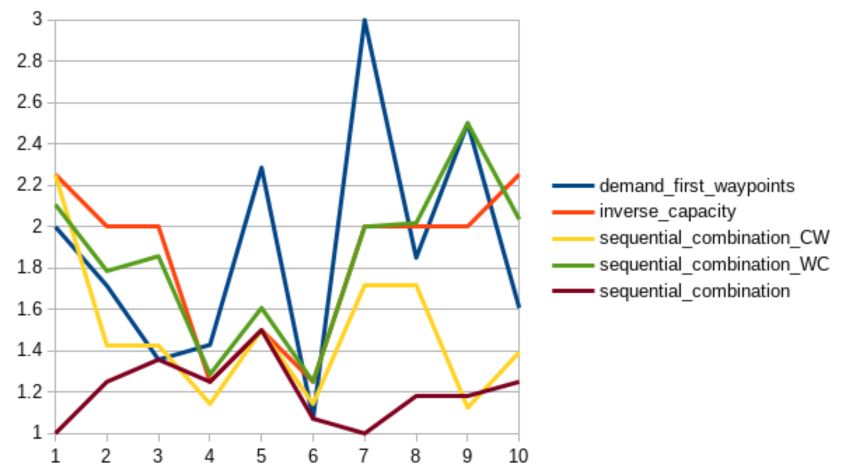
Zweiter Schritt: Ergebnisse

Zielmetrik: MLU (minimum link utilization)

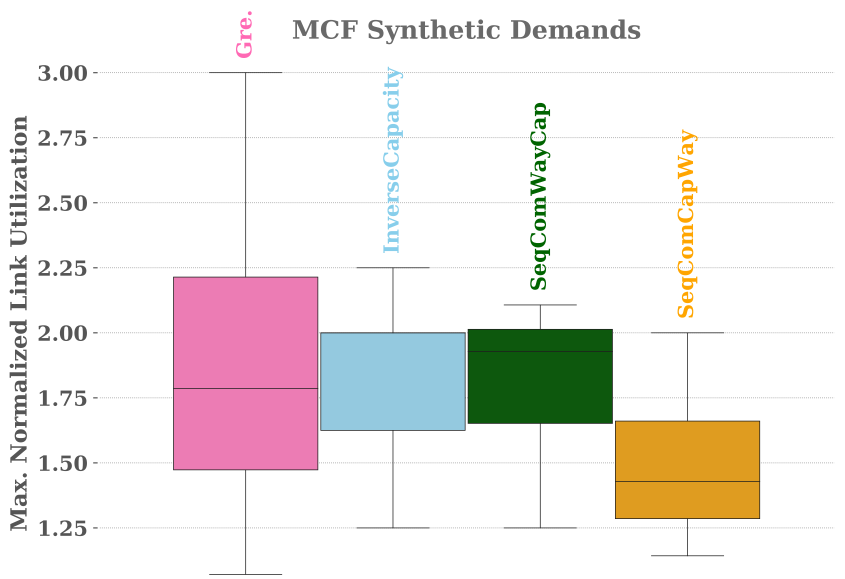


Zweiter Schritt: Ergebnisse

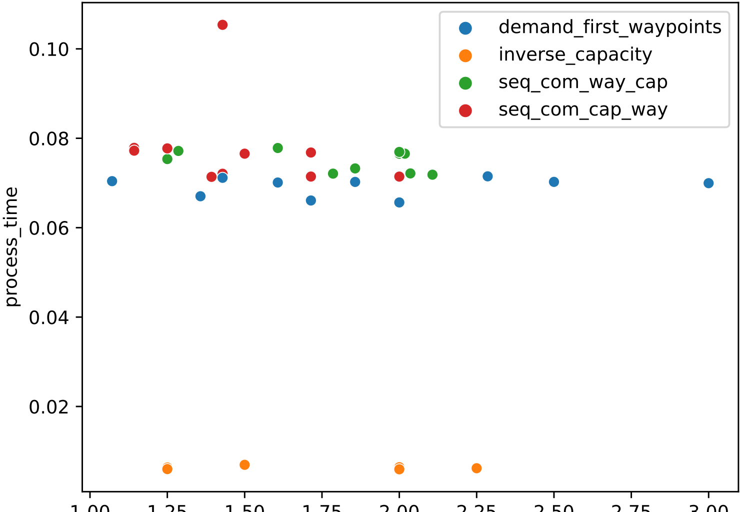
Zielmetrik: MLU (minimum link utilization) + Rechenzeit



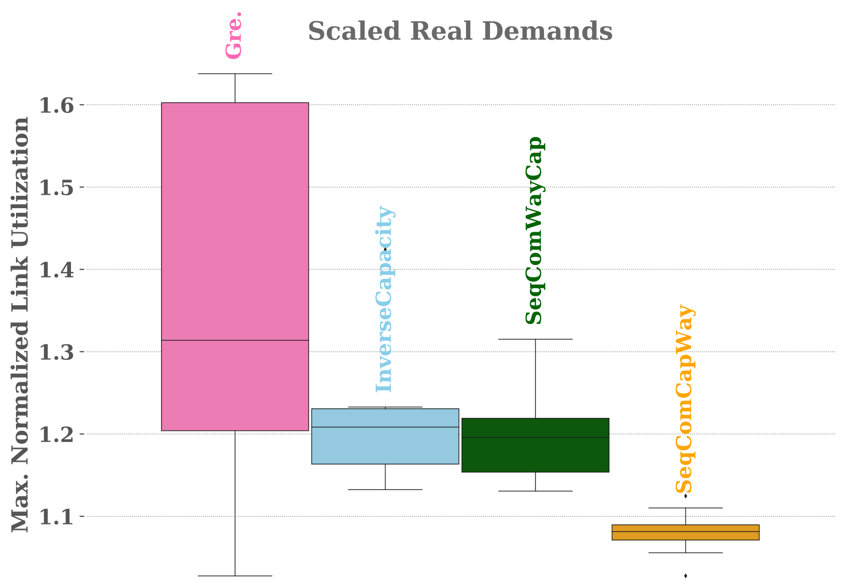
Ergebnisse


MCF Synthetic Demands
Max. Normalized Link Utilization
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
Gre.
InverseCapacity
SeqComWayCap
SeqComCapWay

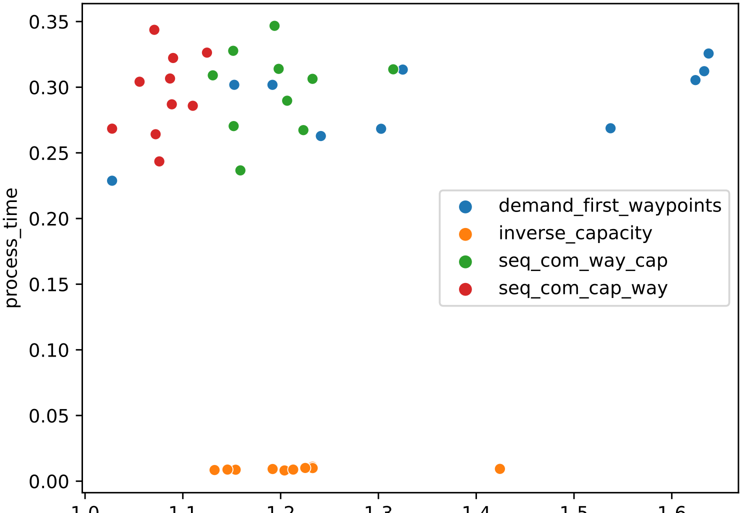
Ergebnisse


demand_first_waypoints
inverse_capacity
seq_com_way_cap
seq_com_cap_way
process_time
0.10
0.08
0.06
0.04
0.02
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00

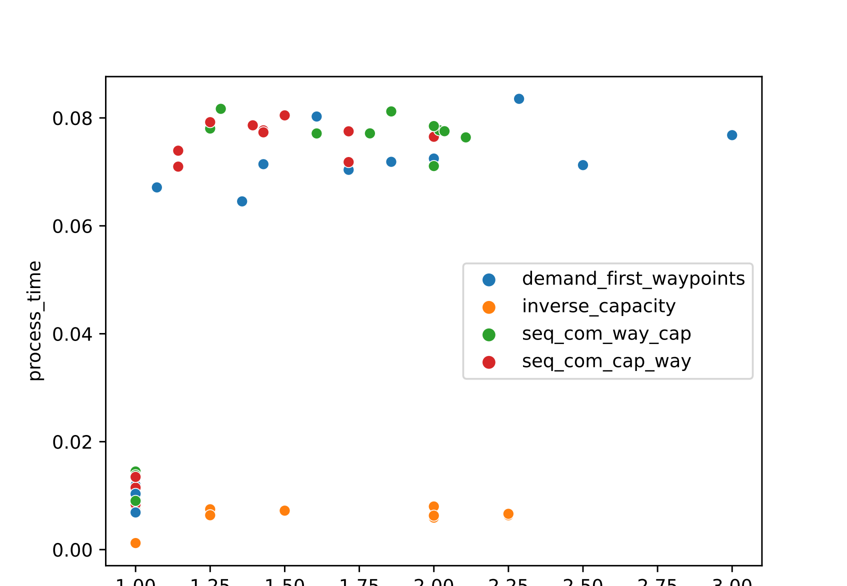
Ergebnisse


Scaled Real Demands
Max. Normalized Link Utilization
1.6
1.5
1.4
1.3
1.2
1.1
Gre.
InverseCapacity
SeqComWayCap
SeqComCapWay

Ergebnisse


process_time
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1.0
1.1
1.2
1.3
1.4
1.5
1.6
● demand_first_waypoints
● inverse_capacity
● seq_com_way_cap
● seq_com_cap_way

Ergebnisse


process_time
0.08
0.06
0.04
0.02
0.00
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
● demand_first_waypoints
● inverse_capacity
● seq_com_way_cap
● seq_com_cap_way

Zweiter Schritt: Fragen

- warum ist SQ_{CW} besser als SQ_{WC}?
- Rechenaufwand von beiden gleich?
- warum SQ so viel besser als SQ_{CW} oder SQ_{WC}?

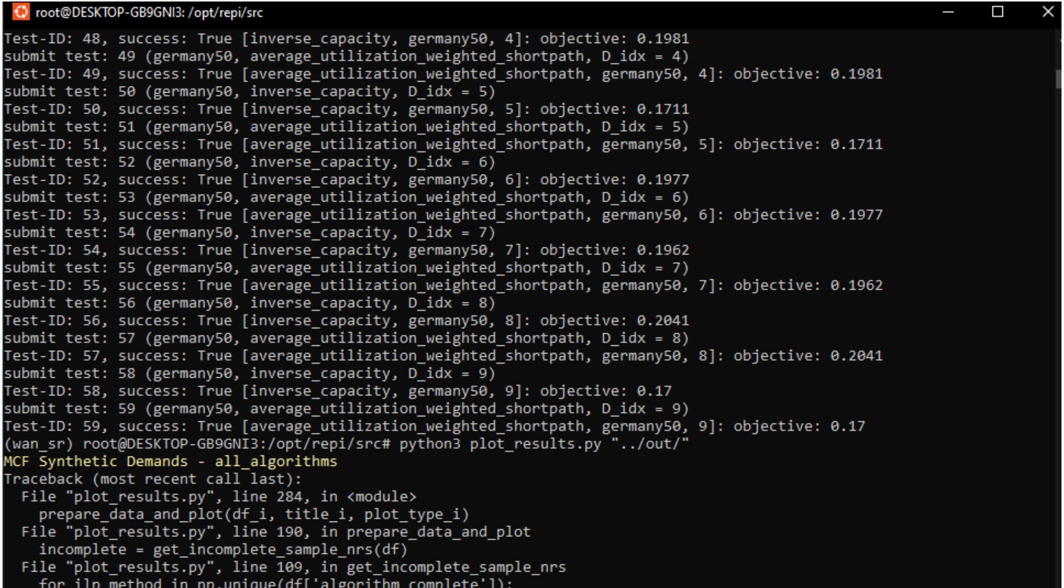
Nächste Schritte:

- Algos genauer analysieren Schritt 3
- Experimente reproduzierbar machen Schritt 4

Fragen oder Anmerkungen?

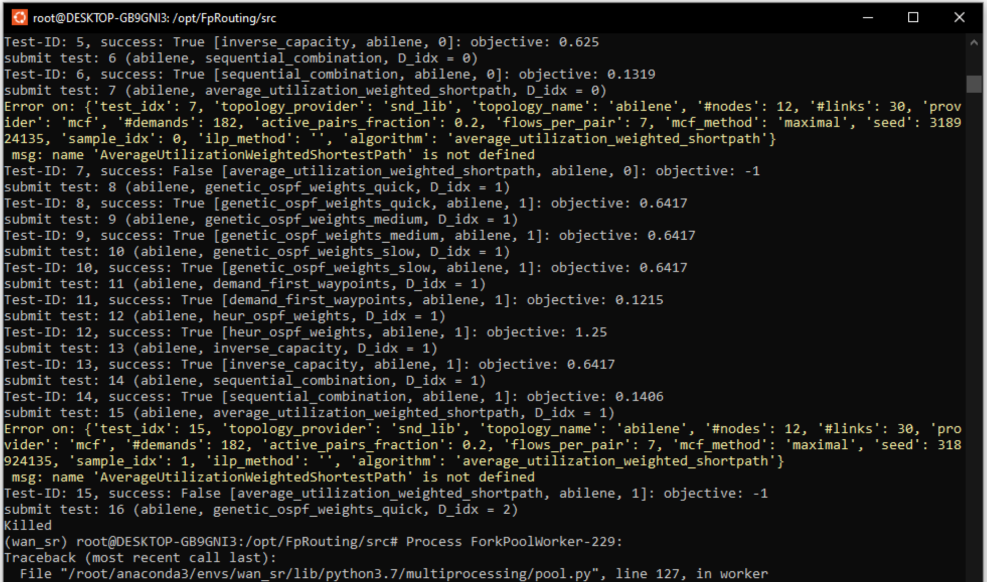
Ergebnisse Reproduktion von Gruppe 3

Frame Title



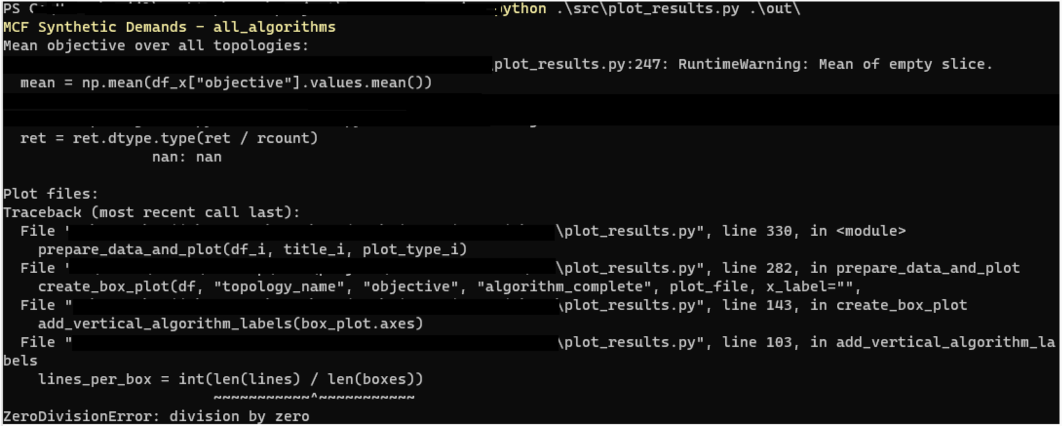
```
Test-ID: 48, success: True [inverse_capacity, germany50, 4]: objective: 0.1981
submit test: 49 (germany50, average_utilization_weighted_shortpath, D_idx = 4)
Test-ID: 49, success: True [average_utilization_weighted_shortpath, germany50, 4]: objective: 0.1981
submit test: 50 (germany50, inverse_capacity, D_idx = 5)
Test-ID: 50, success: True [inverse_capacity, germany50, 5]: objective: 0.1711
submit test: 51 (germany50, average_utilization_weighted_shortpath, D_idx = 5)
Test-ID: 51, success: True [average_utilization_weighted_shortpath, germany50, 5]: objective: 0.1711
submit test: 52 (germany50, inverse_capacity, D_idx = 6)
Test-ID: 52, success: True [inverse_capacity, germany50, 6]: objective: 0.1977
submit test: 53 (germany50, average_utilization_weighted_shortpath, D_idx = 6)
Test-ID: 53, success: True [average_utilization_weighted_shortpath, germany50, 6]: objective: 0.1977
submit test: 54 (germany50, inverse_capacity, D_idx = 7)
Test-ID: 54, success: True [inverse_capacity, germany50, 7]: objective: 0.1962
submit test: 55 (germany50, average_utilization_weighted_shortpath, D_idx = 7)
Test-ID: 55, success: True [average_utilization_weighted_shortpath, germany50, 7]: objective: 0.1962
submit test: 56 (germany50, inverse_capacity, D_idx = 8)
Test-ID: 56, success: True [inverse_capacity, germany50, 8]: objective: 0.2041
submit test: 57 (germany50, average_utilization_weighted_shortpath, D_idx = 8)
Test-ID: 57, success: True [average_utilization_weighted_shortpath, germany50, 8]: objective: 0.2041
submit test: 58 (germany50, inverse_capacity, D_idx = 9)
Test-ID: 58, success: True [inverse_capacity, germany50, 9]: objective: 0.17
submit test: 59 (germany50, average_utilization_weighted_shortpath, D_idx = 9)
Test-ID: 59, success: True [average_utilization_weighted_shortpath, germany50, 9]: objective: 0.17
(wan_sr) root@DESKTOP-GB9GNI3:/opt/repi/src# python3 plot_results.py "../out/"
MCF Synthetic Demands - all_algorithms
Traceback (most recent call last):
  File "plot_results.py", line 284, in <module>
    prepare_data_and_plot(df_i, title_i, plot_type_i)
  File "plot_results.py", line 190, in prepare_data_and_plot
    incomplete = get_incomplete_sample_nrs(df)
  File "plot_results.py", line 109, in get_incomplete_sample_nrs
    for ilp_method in np.unique(df['algorithm_complete']):
```

Frame Title



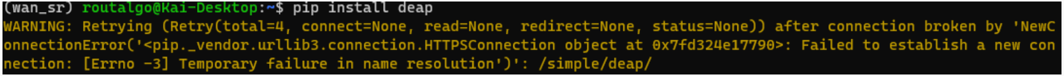
```
Test-ID: 5, success: True [inverse_capacity, abilene, 0]: objective: 0.625
submit test: 6 (abilene, sequential_combination, D_idx = 0)
Test-ID: 6, success: True [sequential_combination, abilene, 0]: objective: 0.1319
submit test: 7 (abilene, average_utilization_weighted_shortpath, D_idx = 0)
Error on: {'test_idx': 7, 'topology_provider': 'snd_lib', 'topology_name': 'abilene', '#nodes': 12, '#links': 30, 'prov
ider': 'mcf', '#demands': 182, 'active_pairs_fraction': 0.2, 'flows_per_pair': 7, 'mcf_method': 'maximal', 'seed': 3189
24135, 'sample_idx': 0, 'ilp_method': '', 'algorithm': 'average_utilization_weighted_shortpath'}
 msg: name 'AverageUtilizationWeightedShortestPath' is not defined
Test-ID: 7, success: False [average_utilization_weighted_shortpath, abilene, 0]: objective: -1
submit test: 8 (abilene, genetic_ospf_weights_quick, D_idx = 1)
Test-ID: 8, success: True [genetic_ospf_weights_quick, abilene, 1]: objective: 0.6417
submit test: 9 (abilene, genetic_ospf_weights_medium, D_idx = 1)
Test-ID: 9, success: True [genetic_ospf_weights_medium, abilene, 1]: objective: 0.6417
submit test: 10 (abilene, genetic_ospf_weights_slow, D_idx = 1)
Test-ID: 10, success: True [genetic_ospf_weights_slow, abilene, 1]: objective: 0.6417
submit test: 11 (abilene, demand_first_waypoints, D_idx = 1)
Test-ID: 11, success: True [demand_first_waypoints, abilene, 1]: objective: 0.1215
submit test: 12 (abilene, heur_ospf_weights, D_idx = 1)
Test-ID: 12, success: True [heur_ospf_weights, abilene, 1]: objective: 1.25
submit test: 13 (abilene, inverse_capacity, D_idx = 1)
Test-ID: 13, success: True [inverse_capacity, abilene, 1]: objective: 0.6417
submit test: 14 (abilene, sequential_combination, D_idx = 1)
Test-ID: 14, success: True [sequential_combination, abilene, 1]: objective: 0.1406
submit test: 15 (abilene, average_utilization_weighted_shortpath, D_idx = 1)
Error on: {'test_idx': 15, 'topology_provider': 'snd_lib', 'topology_name': 'abilene', '#nodes': 12, '#links': 30, 'pro
vider': 'mcf', '#demands': 182, 'active_pairs_fraction': 0.2, 'flows_per_pair': 7, 'mcf_method': 'maximal', 'seed': 318
924135, 'sample_idx': 1, 'ilp_method': '', 'algorithm': 'average_utilization_weighted_shortpath'}
 msg: name 'AverageUtilizationWeightedShortestPath' is not defined
Test-ID: 15, success: False [average_utilization_weighted_shortpath, abilene, 1]: objective: -1
submit test: 16 (abilene, genetic_ospf_weights_quick, D_idx = 2)
Killed
(wan_sr) root@DESKTOP-GB9GNI3:/opt/FpRouting/src# Process ForkPoolWorker-229:
Traceback (most recent call last):
  File "/root/anaconda3/envs/wan_sr/lib/python3.7/multiprocessing/pool.py", line 127, in worker
```

Frame Title



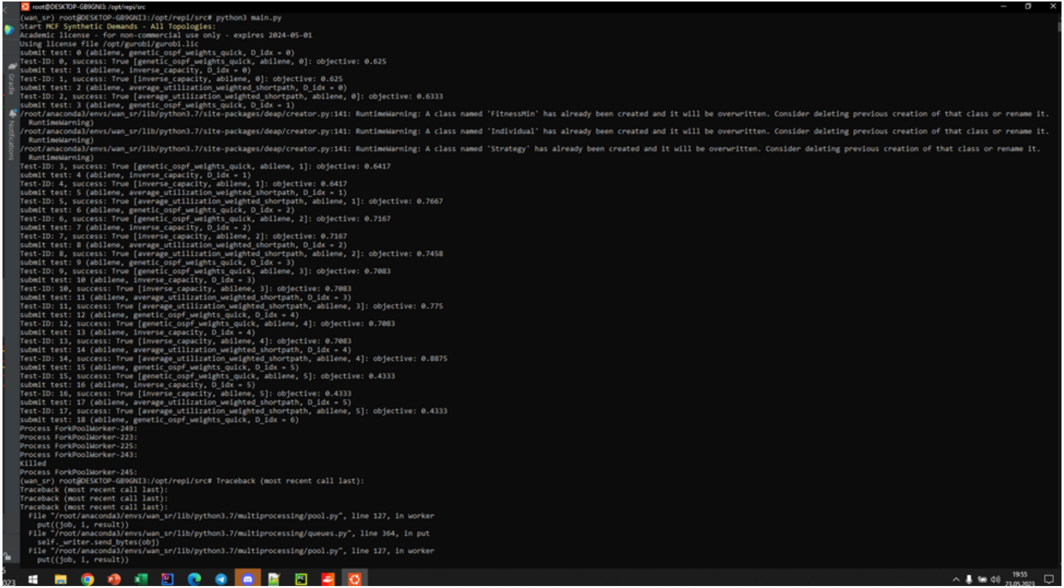
```
PS C[redacted]ython .\src\plot_results.py .\out\
MCF Synthetic Demands - all_algorithms
Mean objective over all topologies:
[redacted]plot_results.py:247: RuntimeWarning: Mean of empty slice.
  mean = np.mean(df_x["objective"].values.mean())
[redacted]
  ret = ret.dtype.type(ret / rcount)
               nan: nan

Plot files:
Traceback (most recent call last):
  File '[redacted]\plot_results.py", line 330, in <module>
    prepare_data_and_plot(df_i, title_i, plot_type_i)
  File '[redacted]\plot_results.py", line 282, in prepare_data_and_plot
    create_box_plot(df, "topology_name", "objective", "algorithm_complete", plot_file, x_label="",
  File "[redacted]\plot_results.py", line 143, in create_box_plot
    add_vertical_algorithm_labels(box_plot.axes)
  File "[redacted]\plot_results.py", line 103, in add_vertical_algorithm_la
bels
    lines_per_box = int(len(lines) / len(boxes))
                        ~~~~~~~~~~~^~~~~~~~~~~~
ZeroDivisionError: division by zero
```

Frame Title



```
(wan_sr) routalgo@Kai-Desktop:~$ pip install deap
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewC
onnectionError('<pip._vendor.urllib3.connection.HTTPSConnection object at 0x7fd324e17790>: Failed to establish a new con
nection: [Errno -3] Temporary failure in name resolution')': /simple/deap/
```

Frame Title



```
(wan_sr) root@DESKTOP-GB9GNI3:/opt/repi/src# python3 main.py
Start MCF Synthetic Demands - All Topologies:
Academic license - for non-commercial use only - expires 2024-05-01
Using license file /opt/gurobi/gurobi.lic
submit test: 0 (abilene, genetic_ospf_weights_quick, D_idx = 0)
Test-ID: 0, success: True [genetic_ospf_weights_quick, abilene, 0]: objective: 0.625
submit test: 1 (abilene, inverse_capacity, D_idx = 0)
Test-ID: 1, success: True [inverse_capacity, abilene, 0]: objective: 0.625
submit test: 2 (abilene, average_utilization_weighted_shortpath, D_idx = 0)
Test-ID: 2, success: True [average_utilization_weighted_shortpath, abilene, 0]: objective: 0.6333
submit test: 3 (abilene, genetic_ospf_weights_quick, D_idx = 1)
/root/anaconda3/envs/wan_sr/lib/python3.7/site-packages/deap/creator.py:141: RuntimeWarning: A class named 'FitnessMin' has already been created and it will be overwritten. Consider deleting previous creation of that class or rename it.
  RuntimeWarning)
/root/anaconda3/envs/wan_sr/lib/python3.7/site-packages/deap/creator.py:141: RuntimeWarning: A class named 'Individual' has already been created and it will be overwritten. Consider deleting previous creation of that class or rename it.
  RuntimeWarning)
/root/anaconda3/envs/wan_sr/lib/python3.7/site-packages/deap/creator.py:141: RuntimeWarning: A class named 'Strategy' has already been created and it will be overwritten. Consider deleting previous creation of that class or rename it.
  RuntimeWarning)
Test-ID: 3, success: True [genetic_ospf_weights_quick, abilene, 1]: objective: 0.6417
submit test: 4 (abilene, inverse_capacity, D_idx = 1)
Test-ID: 4, success: True [inverse_capacity, abilene, 1]: objective: 0.6417
submit test: 5 (abilene, average_utilization_weighted_shortpath, D_idx = 1)
Test-ID: 5, success: True [average_utilization_weighted_shortpath, abilene, 1]: objective: 0.7667
submit test: 6 (abilene, genetic_ospf_weights_quick, D_idx = 2)
Test-ID: 6, success: True [genetic_ospf_weights_quick, abilene, 2]: objective: 0.7167
submit test: 7 (abilene, inverse_capacity, D_idx = 2)
Test-ID: 7, success: True [inverse_capacity, abilene, 2]: objective: 0.7167
submit test: 8 (abilene, average_utilization_weighted_shortpath, D_idx = 2)
Test-ID: 8, success: True [average_utilization_weighted_shortpath, abilene, 2]: objective: 0.7458
submit test: 9 (abilene, genetic_ospf_weights_quick, D_idx = 3)
Test-ID: 9, success: True [genetic_ospf_weights_quick, abilene, 3]: objective: 0.7083
submit test: 10 (abilene, inverse_capacity, D_idx = 3)
Test-ID: 10, success: True [inverse_capacity, abilene, 3]: objective: 0.7083
submit test: 11 (abilene, average_utilization_weighted_shortpath, D_idx = 3)
Test-ID: 11, success: True [average_utilization_weighted_shortpath, abilene, 3]: objective: 0.775
submit test: 12 (abilene, genetic_ospf_weights_quick, D_idx = 4)
Test-ID: 12, success: True [genetic_ospf_weights_quick, abilene, 4]: objective: 0.7083
submit test: 13 (abilene, inverse_capacity, D_idx = 4)
Test-ID: 13, success: True [inverse_capacity, abilene, 4]: objective: 0.7083
submit test: 14 (abilene, average_utilization_weighted_shortpath, D_idx = 4)
Test-ID: 14, success: True [average_utilization_weighted_shortpath, abilene, 4]: objective: 0.8875
submit test: 15 (abilene, genetic_ospf_weights_quick, D_idx = 5)
Test-ID: 15, success: True [genetic_ospf_weights_quick, abilene, 5]: objective: 0.4333
submit test: 16 (abilene, inverse_capacity, D_idx = 5)
Test-ID: 16, success: True [inverse_capacity, abilene, 5]: objective: 0.4333
submit test: 17 (abilene, average_utilization_weighted_shortpath, D_idx = 5)
Test-ID: 17, success: True [average_utilization_weighted_shortpath, abilene, 5]: objective: 0.4333
submit test: 18 (abilene, genetic_ospf_weights_quick, D_idx = 6)
Process ForkPoolWorker-249:
Process ForkPoolWorker-223:
Process ForkPoolWorker-225:
Process ForkPoolWorker-243:
Killed
Process ForkPoolWorker-245:
(wan_sr) root@DESKTOP-GB9GNI3:/opt/repi/src# Traceback (most recent call last):
Traceback (most recent call last):
Traceback (most recent call last):
Traceback (most recent call last):
  File "/root/anaconda3/envs/wan_sr/lib/python3.7/multiprocessing/pool.py", line 127, in worker
    put((job, i, result))
  File "/root/anaconda3/envs/wan_sr/lib/python3.7/multiprocessing/queues.py", line 364, in put
    self._writer.send_bytes(obj)
  File "/root/anaconda3/envs/wan_sr/lib/python3.7/multiprocessing/pool.py", line 127, in worker
    put((job, i, result))
```