

Routingalgorithmen Report

Yannik Buchner
TU Dortmund
Dortmund, Germany
yannic.buchner@tu-dortmund.de

Simon Buschmann
TU Dortmund
Dortmund, Germany
simon-thomas.buschmann@tu-dortmund.de

Jan Draeger
TU Dortmund
Dortmund, Germany
jan.draeger@tu-dortmund.de

Nikita Podibko
TU Dortmund
Dortmund, Germany
nikita.podibko@tu-dortmund.de

Johannes Heinrich
TU Dortmund
Dortmund, Germany
johannes.heinrich@tu-dortmund.de

Malek Haoues Rhaïem
TU Dortmund
Dortmund, Germany
malek.haoues-rhaïem@tu-dortmund.de



Zusammenfassung

Eine gängige Praxis im Traffic Engineering ist es, die Gewichte der Links im Netzwerk zu optimieren, um dann den Traffic auf den kürzesten Pfaden anhand von ECMP zu routen. Diese Herangehensweise nennt sich Link Weight Optimization (LWO) und bietet die Routing-Grundlage für einen Großteil der Internet Service Provider (ISP). Eine andere Herangehensweise ist es, Demands einzeln zu betrachten und anhand von spezifischen Kriterien sog. Wegpunkte zu setzen. Im Laufe unserer Arbeit haben wir uns mit verschiedenen Möglichkeiten beschäftigt, Waypoint Optimization (WPO) zu realisieren und auf Basis der Arbeit *Traffic engineering with joint link weight and segment optimization* [5] Möglichkeiten herausgearbeitet, WPO und LWO zu kombinieren. Außerdem beschäftigen wir uns mit einem neuen Aspekt, dem priorisierten Routing, und stellen dazu konkrete Algorithmen vor, die im Hinblick verschiedenster Kriterien in ausgiebigen Experimenten analysiert werden.

CCS Concepts

• Networks → Traffic engineering algorithms.

Schlagwörter

traffic engineering, network algorithms, segment routing, priority demands

1 Einleitung

In dieser Arbeit haben wir uns das Paper [5] und die zwei dazugehörigen Repositories [4] und [3] angeschaut. Dabei haben wir versucht, die Ergebnisse zu reproduzieren. Auch haben wir uns drei weitere Algorithmen überlegt und diese in das bestehende Projekt eingearbeitet. Unsere Erweiterungen kann man hier finden [2] und [1]. Innerhalb dieses Reportes stellen wir unsere Ergebnisse bei der Einarbeitung unserer eigenen drei Algorithmen vor. Wir erläutern dabei die Idee der Algorithmen, die jeweiligen Vor- und Nachteile, die Schwierigkeiten, welche wir bei der Einarbeitung in die beiden Projekte hatten, und die Ergebnisse, die wir mit den Algorithmen erzielt haben. Zum Schluss gehen wir noch auf die Reproduktion der Gruppe 4 ein, welche auch zwei eigene Algorithmen entwickelt hat.

2 Projekt 1

Im folgenden Teil werden die Vorgehensweise und die Ergebnisse des ersten Projekts vorgestellt. Dabei stand die Frage im Fokus, wie sich verschiedene Zielfunktionen mithilfe geeigneter Algorithmen möglichst optimal erfüllen lassen

2.1 Algorithmen

Die drei Algorithmen, die wir erstellt haben, sind Least Loaded Link First (LLLF), Randomized Load Aware Path Selection (RLAPS) und Apl Waypoints.

3.1.1 LLLF. Der *Least Loaded Link First* (LLL) Algorithmus ist ein heuristisches Verfahren zur Pfadwahl in Kommunikationsnetzen. Ziel ist es, Anfragen für Datenübertragungen so durch das Netzwerk zu leiten, dass die Auslastung einzelner Kanten möglichst gleichmäßig verteilt wird (Maximum Link Utilisation MLU). Der LLLF-Algorithmus betrachtet die aktuelle Belastung der Netzwerkressourcen. Für jeden Demand wird ein Pfad gesucht, der die aktuelle maximale Auslastung des Netzes minimal hält. Dies geschieht durch eine Abwandlung des Dijkstra-Algorithmus, der die relative Auslastung der Links betrachtet.

- **Fairness:** Es wird vermieden, wiederholt denselben, zu verwenden.
- **Greedy-Heuristik:** Es wird eine lokal optimale Entscheidung für die aktuelle Anfrage, ohne globale Optimierung über alle Demands hinweg, getroffen.
- **Laufzeit:** Der Algorithmus läuft effizient, wobei die Laufzeit ungefähr der von Dijkstra entspricht.
- **Praktische Relevanz:** In großen Telekommunikations- und Datennetzen ist LLLF attraktiv, weil er relativ einfach implementierbar ist.

3.1.2 RLAPS. Der Algorithmus Randomized Load-Aware Path Selection (RLAPS) ist ein heuristisches Routingverfahren, das entwickelt wurde, um Netzwerke gegen lokale Überlastung abzusichern und dabei dennoch kurze Pfade zu berücksichtigen. Anstelle deterministischer Shortest-Path-Strategien, die häufig zur Ausbildung von Hotspots führen, nutzt der Ansatz kontrollierte Randomisierung, um Diversität in der Pfadwahl zu erzeugen und dadurch den Datenverkehr gleichmäßiger im Netz zu verteilen. Die Grundidee besteht darin, für jeden Demand aus eine Menge von k -kürzesten Pfaden drei Kandidaten zufällig zu wählen und mit einer lastbasierten Bewertungsfunktion den besten zu wählen. Sei P ein Pfad bestehend aus Kanten (e_1, \dots, e_m) und d die Demand-Größe. Mit f_e der bisher auf e geplanten Last und c_e der Kapazität gilt die Pfadbewertungsfunktion

$$\text{score}(P, d) = \sum_{e \in P} \frac{f_e + d}{c_e}$$

Diese Wahl zielt direkt darauf ab, die entstehende Auslastung auf den von P genutzten Links zu minimieren.

3.1.3 AplWaypoints. Der *AplWaypoints*-Ansatz setzt gezielt Waypoints, um zwischen durchschnittlicher Pfadlänge (APL) und maximaler Linkauslastung (MLU) auszubalancieren. Optimiert wird die kombinierte Zielfunktion

$$J(\lambda) = \lambda \cdot \text{APL} + (1 - \lambda) \cdot \text{MLU}, \quad \lambda \in [0, 1],$$

wobei APL die nach Demand gewichtete Hop-Distanz und MLU die maximale Auslastung (Flow/Kapazität) eines Links ist. Kleine λ priorisieren Robustheit/MLU, große λ kurze Pfade/Latenz.

- **Kandidatenwahl:** Top- k Knoten mit $k = \max(10, \lfloor 0.2|V| \rfloor)$ nach Score $\text{Score}(v) = \alpha \text{BC}(v) + \beta \text{Deg}(v) + \theta \text{DV}(v)$, mit Betweenness BC , Grad Deg und Demand-Volumen DV (Summe der an v anliegenden Demands) und mit Gewichten $\alpha = 0.5$, $\beta = 0.3$, $\theta = 0.15$ ($\alpha + \beta + \theta = 1$).
- **Vorbereitung:** BFS speichert Distanzen und Pfade für (s, t) , (s, v) , (v, t) (gerichteter, ungewichteter Graph).

- **Bewertung/Update:** Für jeden Demand (s, t, d) (absteigend nach d) werden Kandidaten $v \neq s, t$ getestet. Neuer Pfad $s \rightarrow v \rightarrow t$; Flüsse werden entlang alter/ neuer Pfade ab- bzw. zugerechnet, MLU neu bestimmt. APL wird per

$$\Delta \text{APL} = \frac{(\text{dist}(s, v) + \text{dist}(v, t) - \text{dist}(s, t)) \cdot d}{\sum_i d_i}$$

in $O(1)$ aktualisiert. Gewählt wird die Option mit minimalem $J(\lambda)$, sonst bleibt $s \rightarrow t$ direkt.

2.2 Experimente

3.2.1 LLLF. Im Rahmen der Experimente wurde der LLLF-Algorithmus, wie oben beschrieben, in das Projekt integriert. Auch wurden noch weitere kleine Verbesserungen an dem Algorithmus vorgenommen, wie die Demands zu sortieren oder die Verteilung der Demands mehrmals hintereinander durchzuführen. Da der LLLF-Algorithmus, wie die schon existierenden Algorithmen, die MLU optimiert, gab es bei dem Vergleich mit den anderen Algorithmen keinen weiten Aufwand.

3.2.2 RLAPS. Im Rahmen der Experimente wurde der Algorithmus Randomized Load-Aware Path Selection (RLAPS) in die bestehende Testumgebung integriert. Die Implementierung selbst erwies sich vergleichsweise unkompliziert, da der Algorithmus im Kern nur auf der Berechnung von k -kürzesten Pfaden sowie einer einfachen Bewertungsfunktion basiert. Das Ziel der Experimente bestand nicht allein darin, die maximale Linkauslastung (MLU) zu reduzieren, sondern auch die durchschnittliche Linkauslastung (ALU) zu verbessern. Durch die randomisierte Auswahl und die lastbewusste Pfadbewertung sollte der Algorithmus eine gleichmäßigere Lastverteilung im Netzwerk ermöglichen.

3.2.3 AplWaypoints. Als Darstellung der Fähigkeiten des APL-Waypoints Algorithmus sowie dem allgemeinen Vergleich wurden die Algorithmen LLLF, RLAPS, InverseCapacity und GreedyWaypoints im Rahmen von Experiment 1 herangezogen. Dafür wurden alle relevanten Algorithmen sowie eine Vielzahl an unterschiedlichen und bekannten Topologien innerhalb einer Testumgebung implementiert. Zum Vergleich der Algorithmen wurden die Metriken der Laufzeit, der MLU-Auslastung, der ALU-Auslastung und der APL betrachtet. In den Abbildungen 1 werden jeweils für jede Topologie die Gruppe an Algorithmen unter Betrachtung einer ausgewählten Metrik dargestellt. Zur Darstellung wird ein Boxplot für jeden Algorithmus verwendet.

2.3 Resultate

Alle Algorithmen konnten gut in dem Projekt umgesetzt werden und haben sich gut in die schon vorhandenen Algorithmen integriert. In Abbildung 1 sind die Ergebnisse der Algorithmen. Dabei haben alle Algorithmen durchschnittlich gut bei der MLU abgeschnitten, wobei bei einigen Topologien die einen Algorithmen besser waren und bei anderen Topologien die anderen Algorithmen.

3.2.1 LLLF. Die Auswertung der Experimente erfolgte anhand der Metriken Maximal Normalized Link Utilization (MLU) sowie der Ausführungszeit. Die Ergebnisse sind in den Abbildungen aus den Experimenten dargestellt. Man sieht in Abbildung 1a, wie der

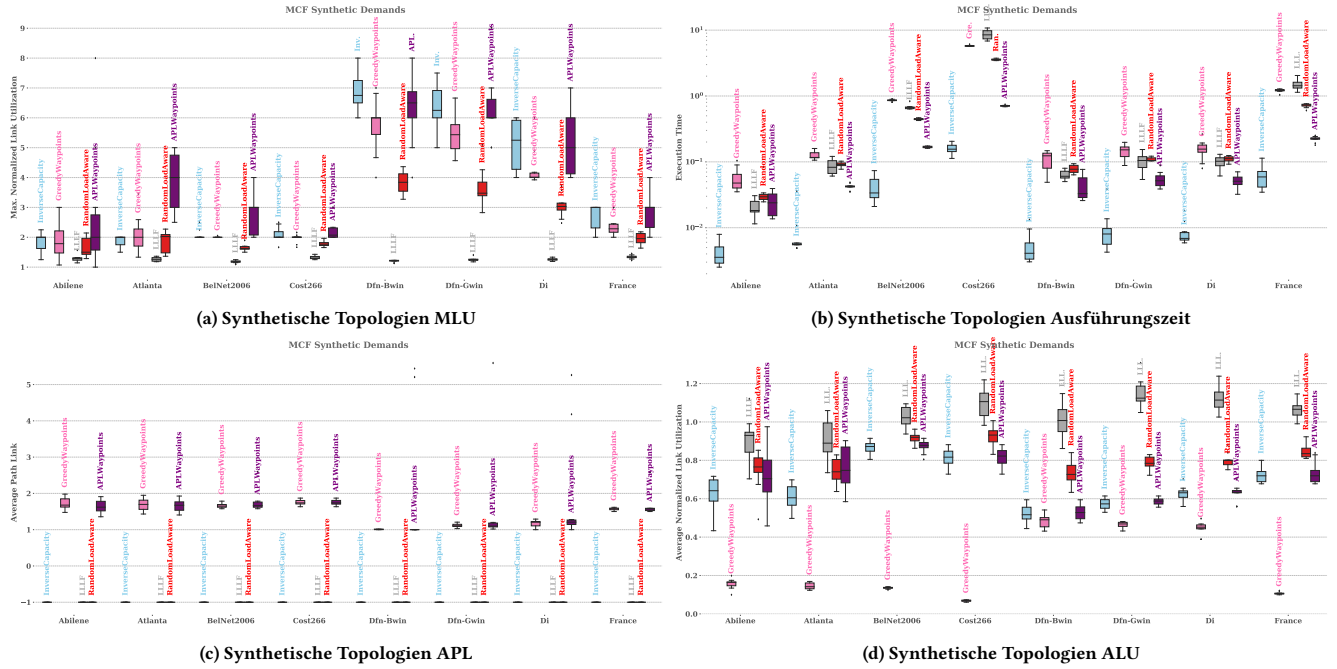


Abbildung 1: Ergebnisse Projekt 1

Algorithmus bei der MLU gut mit den anderen Algorithmen mithalten kann und auch teilweise deutlich besser abschneidet. Diese besonders guten Topologien sind dabei alle sehr dicht, das bedeutet, sie haben viele Kanten und somit auch viele mögliche Pfade. Dadurch hat LLLF eine sehr große Auswahl an möglichen Pfaden und kann die Demands sehr gut auf der Topologie verteilen. Auch sieht man in Abbildung 1b sehr gut, wie er durchschnittliche genauso schnell läuft, wie die schon vorhandenen Algorithmen.

3.2.2 RLAPS. Die Auswertung der Experimente erfolgte anhand der Metriken Average Normalized Link Utilization (ALU), Maximal Normalized Link Utilization (MLU) sowie der Ausführungszeit. Die Ergebnisse sind in den Abbildungen aus den Experimenten dargestellt. Betrachtet man die maximale Linkauslastung (MLU), so wird in Abbildung 1a deutlich, dass RLAPS in nahezu allen getesteten Topologien eine Verringerung von Hotspots erreicht. Während einfache Verfahren wie Inverse Capacity oder Greedy Waypoints oft hohe Spitzenlasten verursachen, gelingt es RLAPS durch die randomisierte Pfadwahl und lastbasierte Bewertung, die maximale Auslastung einzelner Links abzufedern. Ein weiterer Aspekt betrifft die Ausführungszeit, wie in Abbildung 1b dargestellt. Hier zeigt sich, dass RLAPS zwar deutlich langsamer arbeitet als einfache Verfahren wie Unit Weights, jedoch im Vergleich zu exakten den anderen Ansätzen erheblich schneller ist. Die Implementierung des Algorithmus ist zwar sehr simpel und benötigt lediglich lokale Lastinformationen, die Berechnung von k-kürzesten Pfaden und die wiederholte Bewertung mehrerer Kandidatenpfade führen jedoch zu einem deutlichen Mehraufwand. Besonders auffällig ist, dass die Rechenzeit mit wachsender Topologiegöße und Komplexität exponentiell angestiegen ist, was die praktische Anwendbarkeit in sehr großen Netzwerken einschränken kann. Diese eingeschränkte

Skalierbarkeit stellt damit eine der wesentlichen Limitationen von RLAPS dar.

3.2.3 AplWaypoints. Zur Bewertung der Leistungsfähigkeit des APL-Waypoint Algorithmus wurden die in der Methodik beschriebenen Metriken Laufzeit, MLU-Auslastung, ALU-Auslastung und APL betrachtet. Im Folgenden werden die Ergebnisse bezüglich jeder der einzelnen Metriken evaluiert.

Bezüglich der Laufzeit zeigt unser Algorithmus eine durchschnittlich sehr geringe Laufzeit über alle Topologien auf. Dabei ist die Laufzeit ungefähr vergleichbar mit den anderen Algorithmen dieser Gruppe und wird lediglich vom InverseCapacity Algorithmus geschlagen, siehe Abbildung 1b. Die Varianz der Laufzeiten in den Tests bleibt hierbei gering im Allgemeinen und im Vergleich zu den anderen Algorithmen. In der Metrik der MLU-Auslastung fällt vor allem die hohe Varianz der MLU-Auslastung auf. Diese zeigt sich unter anderem auch bei Betrachtung der Box-Plots für verschiedene Topologien. Der durchschnittliche Wert der MLU-Auslastung ist vergleichbar schlechter als bei den zum Vergleich gewählten Algorithmen, siehe Abbildung 1a. Die Ergebnisse zur Metrik APL, zeigen sehr gute für den APL-Waypoints Algorithmus, siehe 1c. Im Allgemeinen schneidet hier unser Algorithmus am besten ab, jedoch nicht deutlich. Des Weiteren haben wir eine tolerierbare Varianz der Ergebnisse, wobei unsere absoluten Minima in fast allen Topologien das beste Ergebnis abbilden. Zusammenfassend zeigt APL-Waypoints in allen Metriken vergleichbare Ergebnisse und teilweise gute Ergebnisse. Trotzdem ist hervorzuheben, dass die größte Schwäche in der MLU-Auslastung liegt, welche im Allgemeinen die wichtigste Metrik darstellt. Positiv hervorzuheben ist hierbei vor allem die Laufzeit und die Ergebnisse bezüglich der APL.

3 Projekt 2

Im Anschluss wird das zweite Projekt betrachtet. Ziel war es hierbei, die Algorithmen aus Projekt 1 nicht nur in Simulationen zu untersuchen, sondern sie zusätzlich in einer virtuellen Netzwerkumgebung praktisch anzuwenden.

3.1 Experimente

4.1.1 LLLF. Um im zweiten Projekt gute Ergebnisse mit LLLF zu erzielen, haben eine neue sehr dichte Topologie 2a entworfen, auf welcher LLLF sehr gut klarkommen sollte. Dabei haben alle Kanten eine Kapazität von 2 und alle Demands sind 1 groß. Die theoretisch mögliche MLU ist 0, 5.

4.1.2 RLAPS. Im zweiten Experiment wurde die Topologie aus Abbildung 3a entworfen, die gezielt gut geeignet für Randomized Load-Aware Path Selection (RLAPS) ist. Sie bietet eine hohe Pfadvelfalt, wodurch die randomisierte Vorauswahl tatsächlich verschiedene Alternativen berücksichtigen kann. Zentrale Engpässe wurden vermieden, sodass Lasten besser verteilt werden können. Die Kapazitäten wurden konsistent modelliert, um die Bewertungsfunktion nicht zu verzerren und die Netzgröße wurde moderat gehalten, damit die Berechnung der k -kürzesten Pfade handhabbar bleibt für die Testumgebung. Zusätzlich wurden für die Reproduzierbarkeit der Ergebnisse Anpassungen an den Bitraten vorgenommen, sodass die Experimente zügig wiederholt werden können.

4.1.3 AplWaypoints. Für einen realitätsnahen Vergleich werden in Experiment 2 die Algorithmen JOINT, Weights und APL-Waypoints in dem Simulationsframework nanonet betrachtet. Dabei hat jeder Algorithmus eine individuell definierte Topologie sowie eine eigen definiertes Routingverhalten.

Dabei wurde für den APL-Waypoints Algorithmus eine Grid-Topologie ausgewählt, siehe Abbildung 4a. Dieses bietet unserem Algorithmus neben der hohen Pfaddiversität eine Kapazitätssetzung, die Umwege um den zentralen Pfad einen erhöhten Mehrwert geben. Des Weiteren wurde neben der Bitrate die Topologie klein gehalten in der Definition der nötigen Dateien, um die Durchlaufzeit gering zu halten.

3.2 Resultate

4.2.1 LLLF. In Abbildung 2b kann man sehen wie, der Weights-, der Joint- und der LLLF-Algorithmus abgeschnitten haben. Man sieht, dass LLLF zwischen Joint und Weights liegt. Es fällt aber auch auf, dass die eigentliche theoretische MLU von 0, 5 nicht erreicht wurde. Das könnte an einigen Fehlern bei der Simulation liegen, die während der ersten paar Minuten häufiger aufgetreten sind, dann aber verschwanden.

4.2.2 RLAPS. RLAPS wurde direkt mit der Weights-Methode auf der erstellten Topologie verglichen (vgl. Abbildung 3b). Während Weights eine konsistent niedrigere maximale Linkauslastung erzielt und nur geringe Varianz aufweist, zeigt RLAPS ein deutlich breiteres Ergebnisintervall. Der Median der MLU liegt bei RLAPS höher, was auf tendenziell schlechtere Ergebnisse hindeutet. Gleichzeitig belegen die Ausreißer nach unten, dass RLAPS in einzelnen Durchläufen durchaus konkurrenzfähige Resultate liefern kann. Die Randomisierung schafft Diversität in der Pfadwahl und kann

Hotspots wirksam vermeiden, führt aber auch zu höherer Ergebnisvarianz. Während deterministische Verfahren wie Weights eine stabile, aber weniger flexible Lösung bieten, bewegt sich RLAPS zwischen sehr guten und weniger guten Resultaten. Für praktische Anwendungen bedeutet dies, dass RLAPS eher als explorativer Ansatz geeignet ist, dessen Leistungsfähigkeit durch mehrere Läufe und Mittelwertbildung abgesichert werden sollte.

4.2.3 AplWaypoints. Der APL-Waypoints Algorithmus zeigt auch im Experiment 2 vergleichbare Ergebnisse. Dabei zeigt Abbildung 4b die MLU in Bezug auf die gegebenen Demands. APL-Waypoints zeigt hier eine deutlich höhere MLU, siehe Abbildung 4b, im Vergleich zu JOINT und Weights. Jedoch zeigt hier unser Algorithmus ein sehr stabiles Ergebnis, gekennzeichnet durch die ausgesprochen niedrige Varianz. Dies unter der hohen Belastung zeigt die Robustheit des Algorithmus. Im Allgemeinen hebt Experiment 2 die schon in Experiment 1 erkannte Schwäche des APL-Waypoint Algorithmus erneut hervor. Die Werte in der Metrik der MLU zeigen einen generellen Nachteil gegenüber den zum Vergleich herangezogenen Algorithmen. Dabei ist positiv lediglich die geringe Varianz und somit die hohe Stabilität der Ergebnisse hervorzuheben. Zusammenfassend schneidet APL-Waypoints, in Bezug auf lediglich die Ergebnisse von Experiment 2, vergleichsweise schlecht ab.

4 Reproduktion

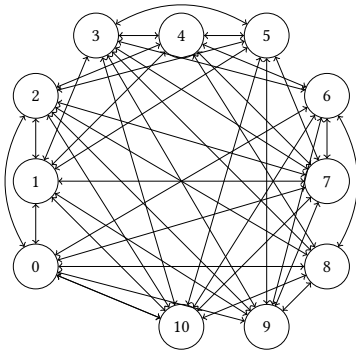
Die Reproduktion der Projekte aus Gruppe 4 verlief insgesamt erfolgreich. Wesentliche Abweichungen von den ursprünglichen Ergebnissen konnten wir nicht feststellen. Bei Projekt 1 war der Code gut dokumentiert und strukturiert. Die Ausführung sowie die grafischen Darstellungen funktionierten ohne Probleme, und die Resultate entsprachen den Originalen. Auch Projekt 2 ließ sich problemlos ausführen. Auffällig war lediglich, dass unsere Ergebnisse etwas besser ausfielen, was vermutlich auf Unterschiede in der Testumgebung zurückzuführen ist. Insgesamt bestätigten unsere Experimente die ursprünglichen Resultate. Die Leistung unserer Algorithmen ordnete sich konsistent in die bestehenden Verfahren ein.

5 Zusammenfassung

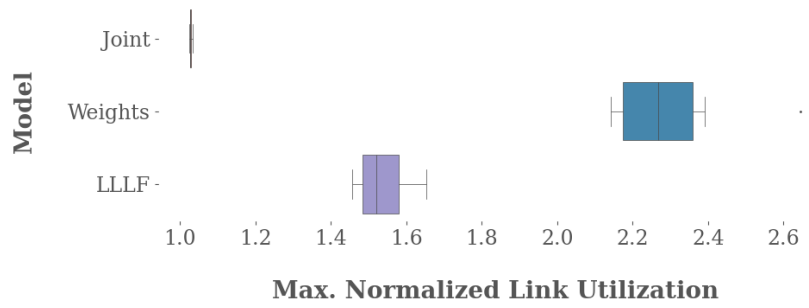
Im Gesamten lief die Einarbeitung unserer Algorithmen gut. Es gab zwar an einigen Stellen Schwierigkeiten, sich in die Projekte einzuarbeiten und unsere Algorithmen passend zu implementieren. Auch hat die Analyse von APL und ALU in den schon implementierten Algorithmen zu weiterem Aufwand geführt, aber diese Schwierigkeiten konnten überwunden werden. Die Ergebnisse der Algorithmen waren dabei gemischt. Teilweise waren sie schneller als die schon existierenden Algorithmen, teilweise aber auch langsamer. Unsere Algorithmen haben sich dabei also gut in die schon existierenden aufgereiht.

Literatur

- [1] 2025. Erweiterung von TE_SR_experiments_2021. https://github.com/Makeandbreak09/TE_SR_experiments_2021. [Online; accessed 11-09-2025].
- [2] 2025. Erweiterung von TE_SR_WAN_simulation. https://github.com/Makeandbreak09/TE_SR_WAN_simulation. [Online; accessed 11-09-2025].

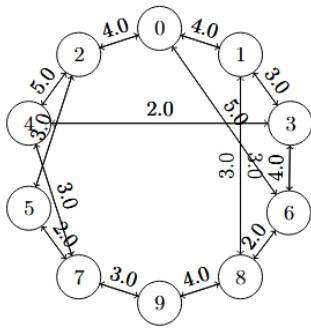


(a) LLLF-Topologie

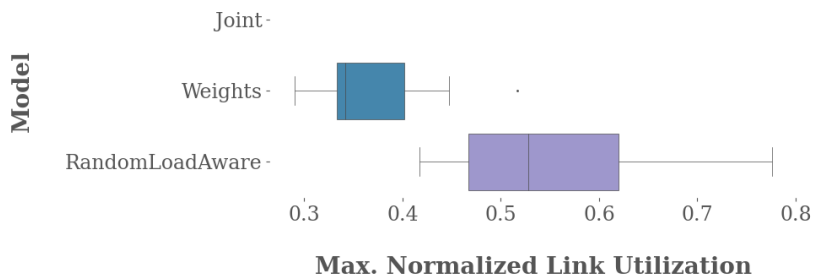


(b) Ergebnisse LLLF Projekt 2

Abbildung 2: Ergebnisse LLLF Projekt 2

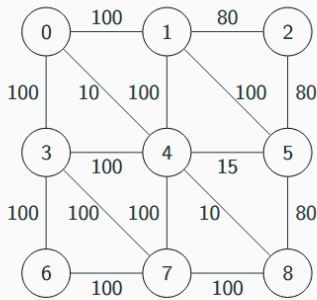


(a) RLAPS-Topologie

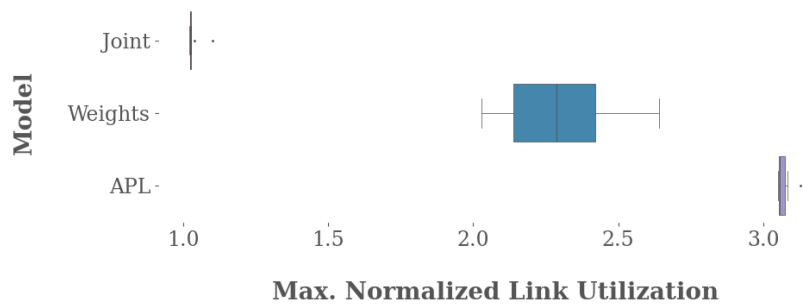


(b) Ergebnisse RLAPS Projekt 2

Abbildung 3: Ergebnisse RLAPS Projekt 2



(a) APL-Waypoints-Topologie



(b) Ergebnisse APL-Waypoints Projekt 2

Abbildung 4: Ergebnisse APL Projekt 2

- [3] Mahmoud Parham, Thomas Fenz, Nikolaus Süß, Klaus-Tycho Foerster, and Stefan Schmid. 2021. TE_SR_experiments_2021. https://github.com/nikolaussuess/TE_SR_experiments_2021. [Online; accessed 11-09-2025].
- [4] Mahmoud Parham, Thomas Fenz, Nikolaus Süß, Klaus-Tycho Foerster, and Stefan Schmid. 2021. TE_SR_WAN_simulation. https://github.com/TFenz/TE_SR_WAN_simulation. [Online; accessed 11-09-2025].

- [5] Mahmoud Parham, Thomas Fenz, Nikolaus Süß, Klaus-Tycho Foerster, and Stefan Schmid. 2021. Traffic engineering with joint link weight and segment optimization. In *Proceedings of the 17th international conference on emerging networking experiments and technologies*. 313–327.