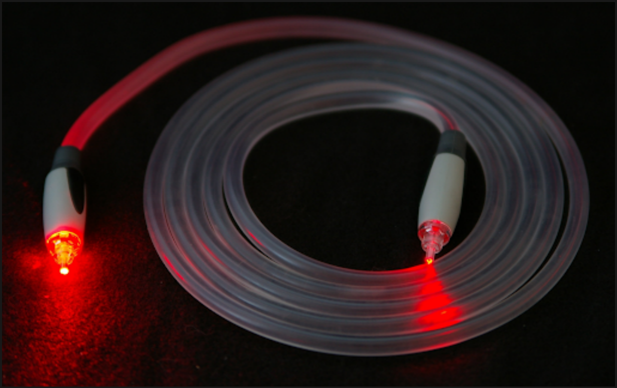
Assessing two-stage Approximation Algorithms for the Routing and Wavelength Assignment Problem

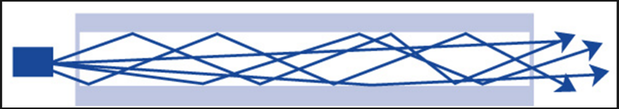
Jesse Hellendoorn

Master Thesis OR
MSc Econometrics & Operations Research
Vrije Universiteit Amsterdam

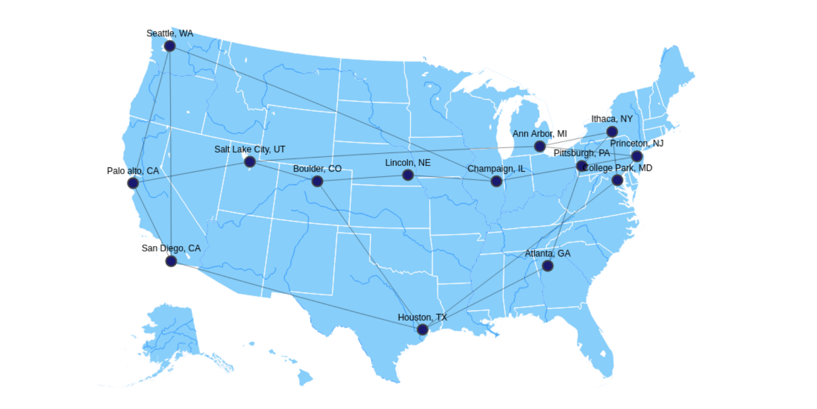
May 25, 2023
GitLab [RL4RWA]

The Routing and Wavelength Assignment Problem

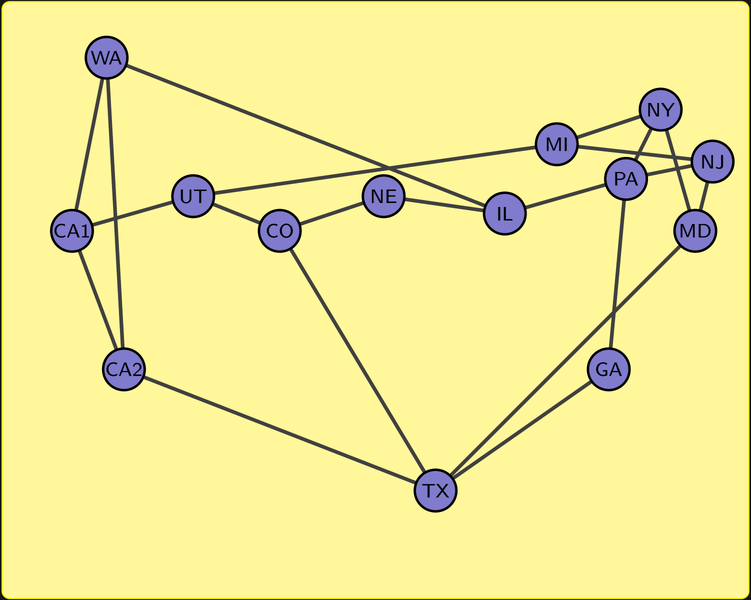




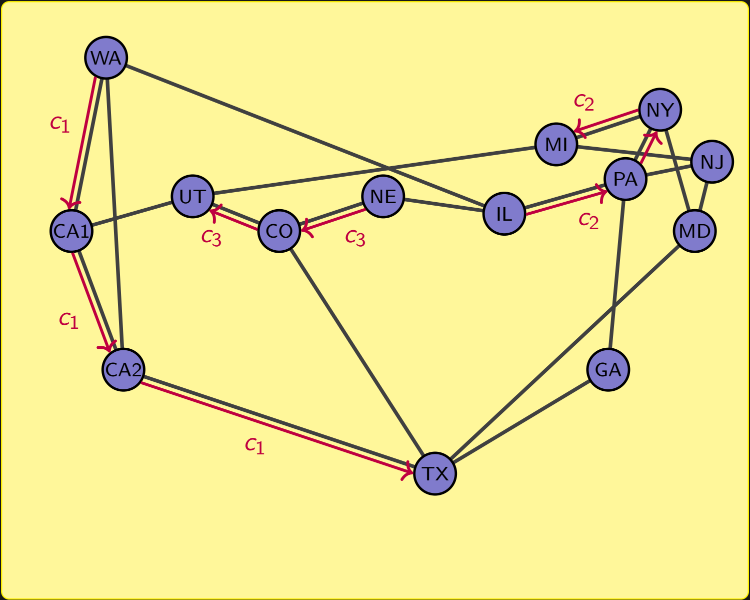
The National Science Foundation Network (NSFNET)



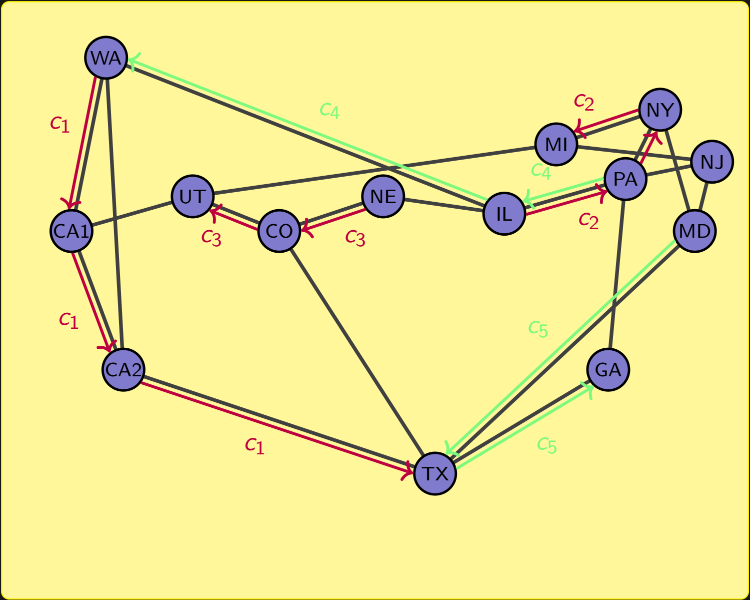
NSFNET



NSFNET

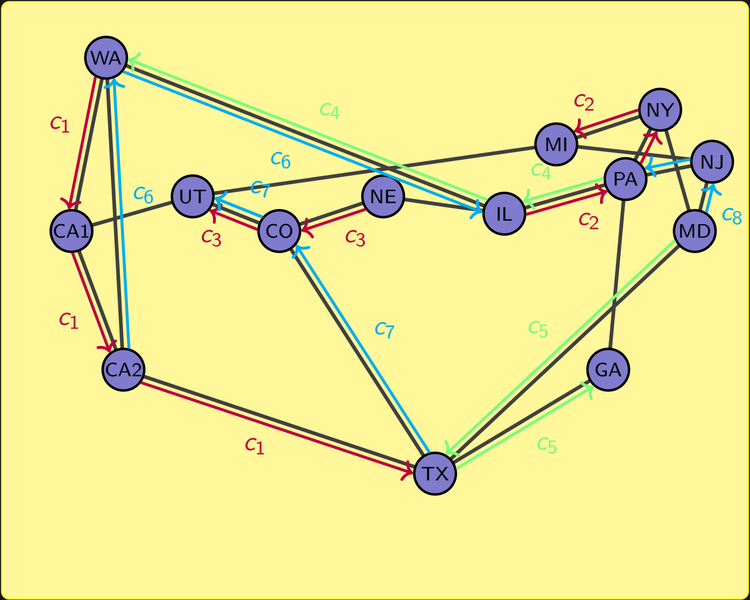


NSFNET



WA
c_1
c_4
c_2
NY
MI
NJ
c_4
PA
UT
NE
CA1
CO
IL
MD
c_3
c_3
c_2
c_1
c_5
CA2
GA
c_1
c_5
TX

NSFNET



WA
CA1
CA2
UT
CO
NE
IL
MI
NY
NJ
PA
MD
GA
TX
c_1
c_1
c_1
c_2
c_2
c_3
c_3
c_4
c_4
c_5
c_5
c_6
c_6
c_7
c_7
c_8

Definition RWA

The Routing and Wavelength Assignment Problem

1. Multicommodity flow problem
2. Distinct wavelength assignment (clash)
3. Wavelength continuity

Definition RWA

Multicommodity Flow Problem

Graph $G = (V, E)$

Definition RWA

Multicommodity Flow Problem

Graph $G = (V, E)$

Demands $C = \{c_1, \ldots, c_n\}$ with $c = (s_c, d_c) \forall c$

Definition RWA

Multicommodity Flow Problem

Graph $G = (V, E)$

Demands $C = \{c_1, \ldots, c_n\}$ with $c = (s_c, d_c) \, \forall c$

Paths $P = \{p_{c_1}^1, \ldots, p_{c_1}^k, \ldots, p_{c_n}^1, \ldots, p_{c_n}^k\}$

Definition RWA

Multicommodity Flow Problem

Graph $G = (V, E)$

Demands $C = \{c_1, \ldots, c_n\}$ with $c = (s_c, d_c) \, \forall c$

Paths $P = \{p^1_{c_1}, \ldots, p^k_{c_1}, \ldots, p^1_{c_n}, \ldots, p^k_{c_n}\}$

Wavelengths $\Lambda = \{\lambda_1, \ldots, \lambda_l\}$

Definition RWA

Multicommodity Flow Problem

Graph $G = (V, E)$

Demands $C = \{c_1, \ldots, c_n\}$ with $c = (s_c, d_c) \forall c$

Paths $P = \{p_{c_1}^1, \ldots, p_{c_1}^k, \ldots, p_{c_n}^1, \ldots, p_{c_n}^k\}$

Wavelengths $\Lambda = \{\lambda_1, \ldots, \lambda_l\}$

Goal: Given above resources, maximize the amount of concurrent demands that can be supported in the network

Definition RWA

Problem Formulation

* Distinct wavelength assignment (clash)

  $\forall\, e \in E$, for each $\lambda \in \Lambda$ at most one path is supported
* Wavelength continuity

  Every p_c with demand c is assigned to a single $\lambda \in \Lambda$ or to ε

Lightpath: a tuple (p_c, λ) comprising a path p_c and a wavelength λ

Definition RWA

Optimization Function

Let $P^+ \subseteq P$ be the set of accepted paths, where a connection c is provisioned by at most a single p_c^+ or not at all

$$f(P^+) = |P^+| + \frac{1}{1 + \sum_{p^+ \in P^+} |p^+|}$$

Then optimize $Z = \max f(P^+)$

Definition RWA

Research Question

"What is the approximation ratio of the best performing reinforcement learning algorithms compared to the best performing non-reinforcement learning algorithms?"

Implementation Model

Model Assumptions

Vertices can be endpoint or transit to multiple connections

Implementation Model

Model Assumptions

Vertices can be endpoint or transit to multiple connections

Edges have unit distance and are symmetric, bidirectional

Implementation Model

Model Assumptions

Vertices can be endpoint or transit to multiple connections

Edges have unit distance and are symmetric, bidirectional

Connections are static, knowable, and continuous without queuing or delaying under the condition that $c_s \neq c_d$

Implementation Model

Model Assumptions

Vertices can be endpoint or transit to multiple connections

Edges have unit distance and are symmetric, bidirectional

Connections are static, knowable, and continuous without queuing or delaying under the condition that $c_s \neq c_d$

Paths $k = 25$

Implementation Model

Model Assumptions

Vertices can be endpoint or transit to multiple connections

Edges have unit distance and are symmetric, bidirectional

Connections are static, knowable, and continuous without queuing or delaying under the condition that $c_s \neq c_d$

Paths $k = 25$

Time Complexity RWA $\in$ NP-Complete and P $\neq$ NP

Implementation Model

Simulation Model Parameters

$|V| = 14, |E| = 21$

Implementation Model

Simulation Model Parameters

$$|V| = 14, |E| = 21$$

$$|\Lambda| = \left\lceil \frac{|V|}{\sqrt{|E|}} \right\rceil = 4$$

Implementation Model

Simulation Model Parameters

$$|V| = 14, |E| = 21$$

$$|\Lambda| = \left\lceil \frac{|V|}{\sqrt{|E|}} \right\rceil = 4$$

$$|C| \sim U\left(\left\lceil \sqrt{|V| \cdot |\Lambda|} \right\rceil, 2\left\lceil \sqrt{|V| \cdot |\Lambda|} \right\rceil\right) = U(15, 30)$$

Implementation Model

Simulation Model Parameters

$$|V| = 14, |E| = 21$$

$$|\Lambda| = \left\lceil \frac{|V|}{\sqrt{|E|}} \right\rceil = 4$$

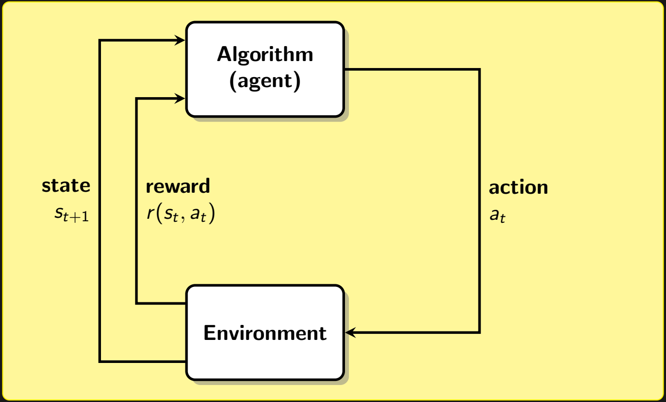
$$|C| \sim U\left(\left\lceil \sqrt{|V| \cdot |\Lambda|} \right\rceil, 2 \left\lceil \sqrt{|V| \cdot |\Lambda|} \right\rceil \right) = U(15, 30)$$

C using random sampling pairs from V

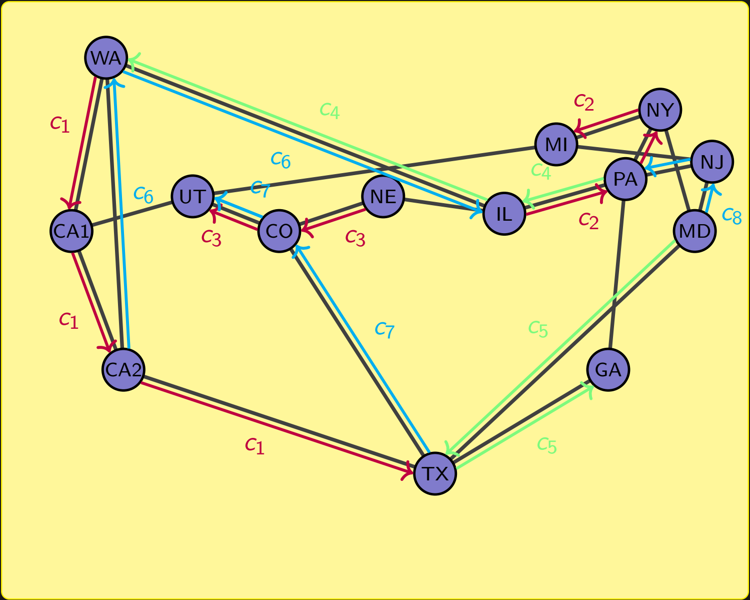
Implementation Model

Residual WA	Assignment of unassigned wavelengths
RL WA Algorithms	Q-Learning algorithm, Deep Q-Learning algorithm, A2C Algorithm, PPO Algorithm
Base WA Algorithms	Random Base, Shortest-Path Base
Routing	Yen's K-shortest Paths Algorithm

Implementation Model



Implementation Model



Implementation Model

State space S

Short: $\{\lambda, \lambda, \lambda, \lambda, \lambda, \lambda, \lambda, \lambda, \varepsilon, \varepsilon\}$ with $\dim(S) = (|\Lambda| + 1)^{|C|}$

Binary: $\{1, 1, 1, 1, 1, 1, 1, 1, 0, 0\}$ with $\dim(S) = 2^{|C|}$

Implementation Model

State space S

Short: $\{\lambda, \lambda, \lambda, \lambda, \lambda, \lambda, \lambda, \lambda, \varepsilon, \varepsilon\}$ with $\dim(S) = (|\Lambda| + 1)^{|C|}$

Binary: $\{1, 1, 1, 1, 1, 1, 1, 1, 0, 0\}$ with $\dim(S) = 2^{|C|}$

Action space A

$\forall\ c \in C$

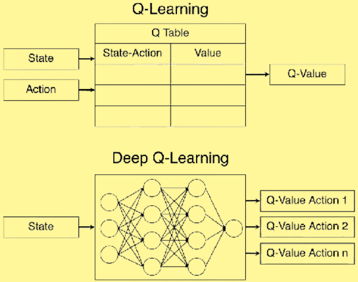
Add path of c opt

Add path of c prob

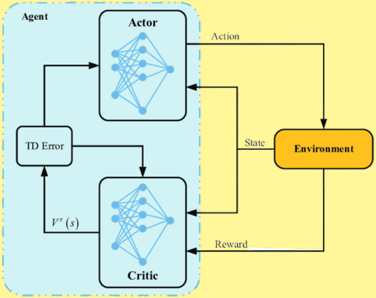
Remove any path of c

with $\dim(A) = 3 \cdot |C|$

Q-learning & Deep Q-learning



Advantage Actor Critic & Proximal Policy Optimization



Other algorithms and the MCF Upper Bound

Multicommodity flow upper bound

$|C| + \frac{1}{1+\sum_{c \in C} |p_c^1|}$

Other algorithms and the MCF Upper Bound

Multicommodity flow upper bound

$|C| + \frac{1}{1+\sum_{c \in C} |p_c^1|}$

Integer Linear Programming

Solving ILP with GEKKO's Advanced Process OPTimizer (APOPT) solver

Other algorithms and the MCF Upper Bound

Multicommodity flow upper bound

$|C| + \frac{1}{1+\sum_{c \in C} |p_c^1|}$

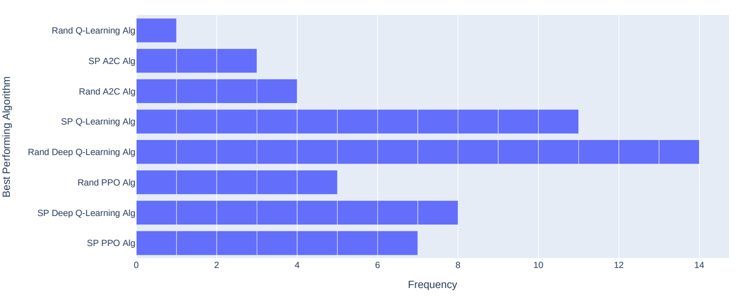
Integer Linear Programming

Solving ILP with GEKKO's Advanced Process OPTimizer (APOPT) solver

Dependency Graph Algorithm

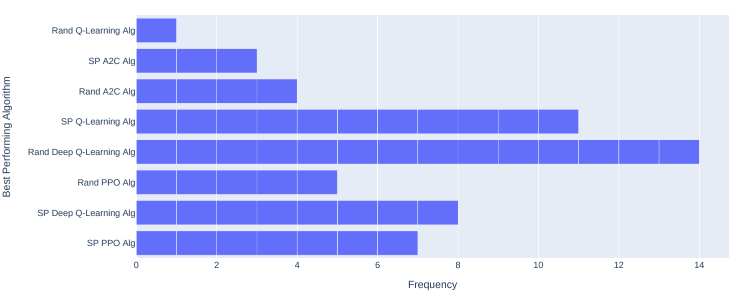
Combination of identifying rewarding paths and random sampling

Results



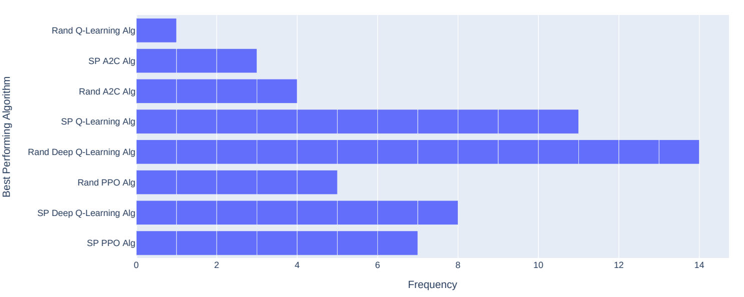
Best Performing Algorithm
Rand Q-Learning Alg
SP A2C Alg
Rand A2C Alg
SP Q-Learning Alg
Rand Deep Q-Learning Alg
Rand PPO Alg
SP Deep Q-Learning Alg
SP PPO Alg
0
2
4
6
8
10
12
14
Frequency

Results



1. SP Q-learning algorithm 18.938
2. SP Deep Q-learning algorithm 18.799
3. Rand Deep Q-learning algorithm 18.779

Results

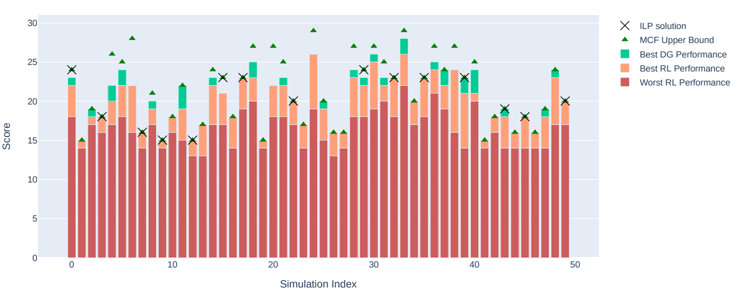


1. SP Q-learning algorithm 18.938
2. SP Deep Q-learning algorithm 18.799
3. Rand Deep Q-learning algorithm 18.779

MCF upper bound 21.382

SP dependency graph algorithm 21.256

Results



Score
Simulation Index
ILP solution
MCF Upper Bound
Best DG Performance
Best RL Performance
Worst RL Performance

Conclusion

Research Question

"What is the approximation ratio of the best performing reinforcement learning algorithms compared to the best performing non-reinforcement learning algorithms?"

Conclusion

Research Question

"What is the approximation ratio of the best performing reinforcement learning algorithms compared to the best performing non-reinforcement learning algorithms?"

Average Approximation Ratios

Best RL to best dependency graph algorithm: 0.984 (0.902)

Best RL to ILP is: 0.974 (0.899)

Conclusion

Research Question

"What is the approximation ratio of the best performing reinforcement learning algorithms compared to the best performing non-reinforcement learning algorithms?"

Average Approximation Ratios

Best RL to best dependency graph algorithm: 0.984 (0.902)

Best RL to ILP is: 0.974 (0.899)

Best Performing RL Algorithms

1. Shortest-path Q-learning Algorithm
2. Shortest-path Deep Q-Learning Algorithm
3. Random Deep Q-Learning Algorithm

Discussion

Reflection on Limitations

* Only considered NSFNET & a single wavelength intensity
* Reliance on base algorithms and residual wavelength assignment, imprecise state representation
* Static perfectly predictable, stable discrete-time connections
* Graph edge uniformity

Thank you for your attention!

Thesis' Contributions

* Created a comprehensive model for RWA benchmarking
* Demonstrated how RWA can be solved using RL
* Introduced methods to assess the algorithmic performance for RWA