Monte Carlo Search Algorithms for Network Traffic Engineering

Chen DANG[1,2], Cristina BAZGAN[2], Tristan CAZENAVE[2], Morgan CHOPIN[1], Pierre-Henri WUILLEMIN[3]

[1]Orange Labs, Châtillon, France
[2]Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, F-75016 Paris, France
[3]Sorbonne Université, CNRS, UMR 7606, LIP6, F-75005 Paris, France

2022 November

Contents

Contents

Problematic

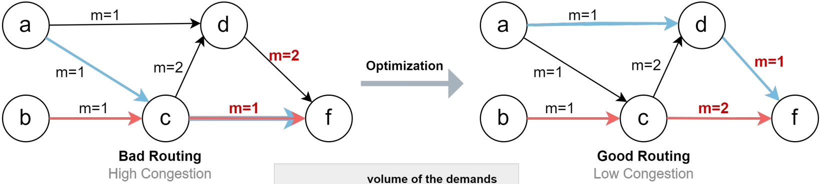
Many telecommunication networks use **shortest paths** protocols (OSPF, IS-IS) for the transportation of packets.

+ Easy to implement
- How to choose the weights to minimize the traffic congestion?

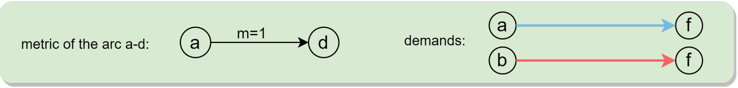
Problematic

Many telecommunication networks use **shortest paths** protocols (OSPF, IS-IS) for the transportation of packets.

+ Easy to implement
- How to choose the weights to minimize the traffic congestion?



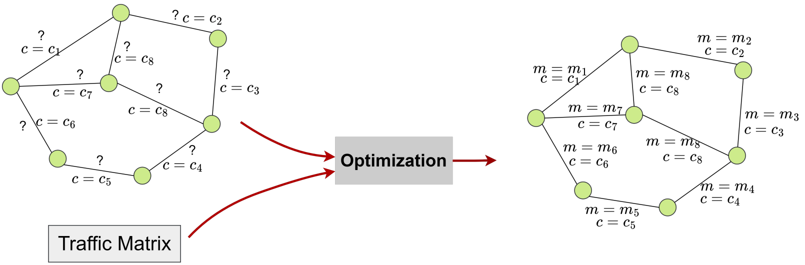
$$\text{Congestion} = \frac{\text{volume of the demands}}{\text{capacity of the arc}}$$



Problematic

Many telecommunication networks use **shortest paths** protocols (OSPF, IS-IS) for the transportation of packets.

+ Easy to implement
- How to choose the weights to minimize the traffic congestion?



Problematic

Many telecommunication networks use **shortest paths** protocols (OSPF, IS-IS) for the transportation of packets.

+ Easy to implement
- How to choose the weights to minimize the traffic congestion?

Formally, we consider the following minimization problem

Min Cong Shortest Path Routing (MIN-CON-SPR)

Input: A graph $G = (V, A)$, a set of terminals $W \subseteq V$, a set $D \subseteq W \times W$ of demands

Output: A set of weights (metrics) M that minimize the maximum congestion over all edges in the resulting routing.

State of the Art

Internet Traffic Engineering by Optimizing OSPF Weights, Bernard Fortz and Mikkel Thorup, 2000

- optimizing the weight settings for a given set of demands is NP-hard

Exact method:

An Integer Programming Algorithm for Routing Optimization in IP Networks, Andreas Bley, 2011

Heuristics:

Local search

Internet Traffic Engineering by Optimizing OSPF Weights, Bernard Fortz and Mikkel Thorup, 2000

Genetic Algorithm

A Hybrid Genetic Algorithm for the Weight Setting Problem in OSPF/IS-IS Routing, Buriol et al, 2005

Contents

Nested Rollout Policy Adaptation

To have faster algorithm that also works on huge graphs,
we use **Nested Rollout Policy Adaptation(NRPA)** to solve the problem.

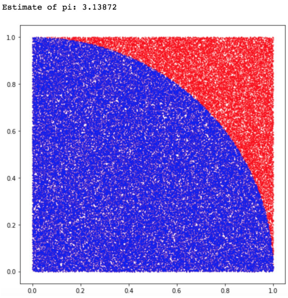
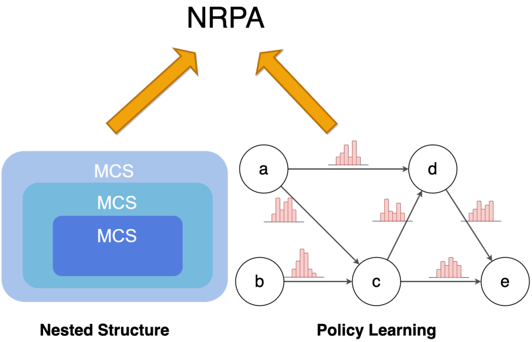
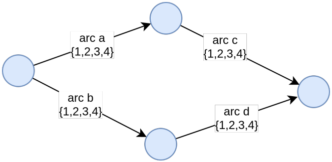


Figure: Monte Carlo: Repeated sampling to obtain numerical results



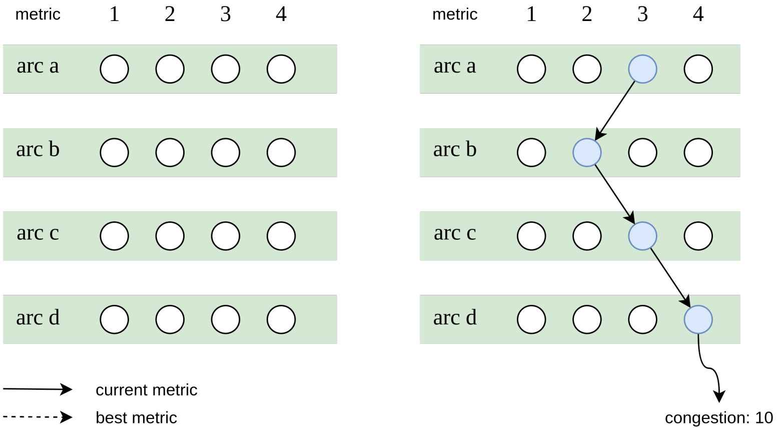
NRPA policy learning illustration



Suppose that we have a graph:

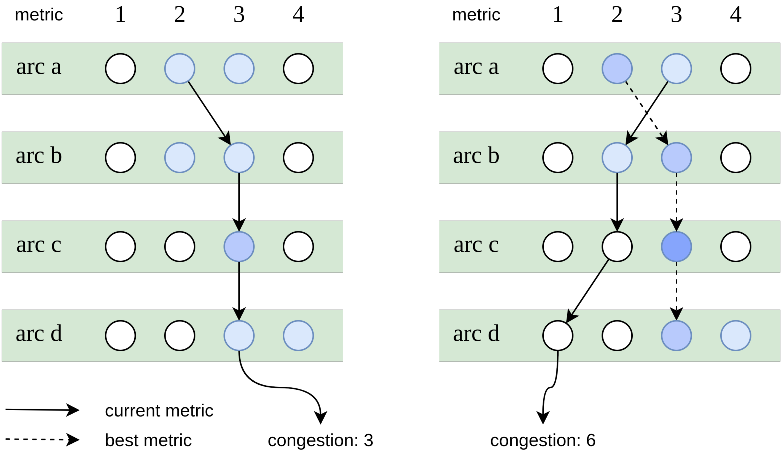
- the arcs are a, b, c, d
- each arc can choose 1, 2, 3 or 4 as the weight

NRPA policy learning illustration



A darker colour means more likely to be selected

NRPA policy learning illustration



A darker colour means more likely to be selected

NRPA Pseudocode

Algorithm NRPA(level, policy)

```
if level = 0 then
    cong, metric = simulation(policy)
    return  cong, metric
else
    for N iterations do
        cong, metric = NRPA(level-1, policy)
        update_policy(BestCong, BestMetrics)
    end for
    return  BestCong, BestMetrics
end if
```

Number of iterations: N

Number of simulations $= N^L$

- choice is made according to a weight w_i

Improvements of NRPA

Considered variant of NRPA

- Stablized NRPA
    - use multiple simulations instead of one
    - more stable
    - more time-consuming

Proposed improvement for NRPA

- Force Exploration
    - Choose a random weight for a random arc if the current solution is already explored
    - Force exploration instead of exploitation
    - Useful for small-medium sized graphs
- Unique Metric
    - Weights on all arcs have different values

Contents

Experiments

Implementation

- implemented in C++

Experimental environment

- Experiments performed on a server with a 64-core Intel(R) Xeon(R) Gold 5218 CPU and 125 GB of memory

Dataset

SNDLib

Welcome to snd·lib
the data source for all people working on optimization of telecommunications networks

Figure: Abilene

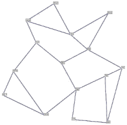


Figure: Atlanta

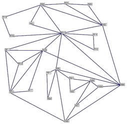
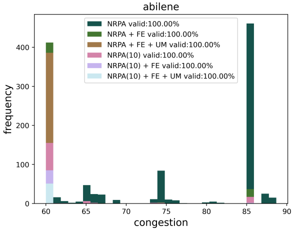


Figure: France

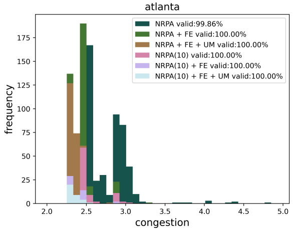
Random Generated Graphs

- random
- waxman

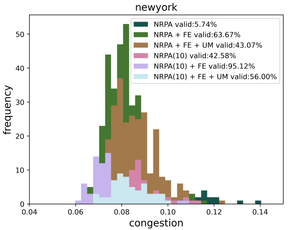
Variants of NRPA



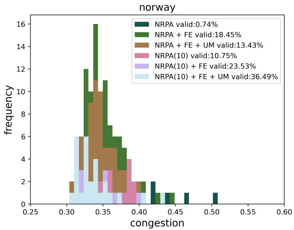
(a) $|M|$=50, t=10 min



(b) $|M|$=50, t=10 min



(c) $|M|$=100, t=30 min



(d) $|M|$=150, t=60 min

Figure: Distribution of the congestion values with all constraints on SNDlib graphs.
FE: Force Exploration, UM: Unique Metric, NRPA(10): Stablized NRPA with 10 simulations

Comparison with other heuristics

Table: Maximum congestion value of state-of-the-art heuristics and NRPA. The value is in **bold** if it is the best one among heuristics. A value is followed by * if equal to the lower bound LPLB. Each graph is tested in a fixed time and each value is averaged on 5 independent executions.

name	$\mid V \mid$	$\mid A \mid$	$\mid K \mid$	Unit OSPF	InvCap OSPF	Local Search	NRPA	LPLB
abilene	12	30	132	187.55	89.48	60.42	**60.412**	60.411
atlanta	15	44	210	3.26	3.37	**2.22**	**2.22**	2.18
france	25	90	300	4.12	4.12	**2.53**	2.56	2.41
nobel-us	14	42	91	37.15	37.15	**24.4**	24.7	24.2
nobel-eu	28	82	378	13.31	13.31	10.68	**10.67***	10.67
brain	161	332	14311	1.415	1.415	0.962	**0.903***	0.903
rand50a	50	132	2450	7.9	7.9	**5.55**	5.77	5.55
rand50b	50	278	2450	**2.88***	**2.88***	**2.88***	**2.88***	2.88
rand100a	100	278	9900	15.71	15.71	10.42	**9.59**	9.35
rand100b	100	534	9900	4.15	4.15	4.38	**3.85**	3.76
wax50a	50	142	2450	6.46	6.46	**4.63**	4.66	4.59
wax50b	50	298	2450	**2.279***	**2.279***	2.284	**2.279***	2.279
wax100a	100	284	9900	17.46	17.46	15.049	**15.048**	15.048
wax100b	100	492	9900	5.51	5.51	4.14	**4.04**	3.44

Comparison with other heuristics

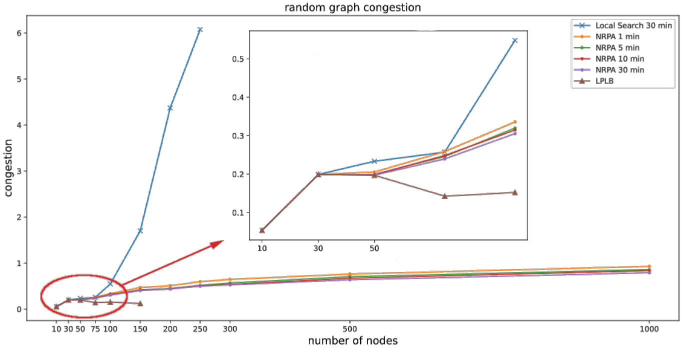


Figure: Congestion with respect to the number of the nodes

Contents

Conclusion

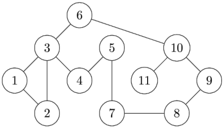
- Model the MIN-CON-SPR problem with Monte Carlo method NRPA
- Propose and improve the NRPA for the problem
- NRPA still has many potential for improvements

Work in Progress: HSPR

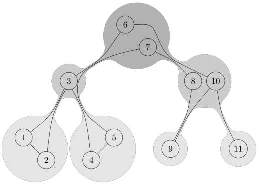
In general, the calculation of the congestion takes more than 90% of the total execution time.

We propose **Hierarchical Shortest Path Routing** to calculate the congestion

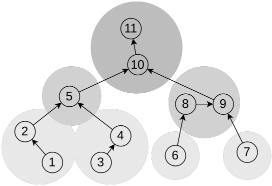
- based on a state-of-the-art approach for shortest path
- very fast calculation for huge telecommunication networks
- network structure is known and fixed, metrics are always changing



(a) Example Graph



(b) Small Balanced Separator



(d) Elimination Tree

Work in Progress: Warm-Starting NRPA

Some NRPA executions can be stuck in a local minimum very quickly, and waste time exploring in a wrong direction

Warm-start NRPA

- reject the executions which don't work well
- continue the **good** executions
- restart often

End

Thank you