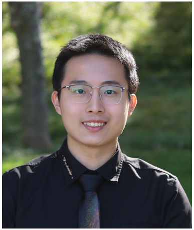
Adversarial Network Optimization under Bandit Feedback: Maximizing Utility in Non-Stationary Multi-Hop Networks

Yan Dai[1←2]   Longbo Huang[2]




[1]ORC, MIT   [2]IIIS, Tsinghua

Stochastic Network Optimization (SNO)

Dynamically allocates resources in networks to fulfill demands, with **strong & rigorous performance guarantees** including

- Throughput Maximization, *e.g.*, [Tassiulas and Ephremides, 1992; 2002; Dai and Lin, 2005; Shah and Shin, 2012], ...
- Delay Minimization, *e.g.*, [Eryilmaz and Srikant, 2007], ...
- Utility Maximization, *e.g.*, [Neely et al., 2005; Georgiadis et al., 2006; Jiang and Walrand, 2009], ...

as well as **numerous successful applications** including

- Wireless Networks [Lin and Shroff, 2006; Srikant and Ying, 2013]
- Cloud Computing [Meng et al., 2010; Maguluri et al., 2012]
- Supply Chain Management [Rahdar et al., 2018]

Key Limitations of Vanilla SNO

Despite huge success, vanilla SNO faces critical limitations:

Limitation 1: Stationary Assumption

- Classical SNO requires network conditions (*e.g.*, arrival / service rates, capacities) to be *stationary over time*
- Fails in reality: auto driving, mobile networks, DDoS, ...
- $\Rightarrow$ Consider **Adversarial Network Optimization (ANO)**

Key Limitations of Vanilla SNO

Despite huge success, vanilla SNO faces critical limitations:

Limitation 1: Stationary Assumption

- Classical SNO requires network conditions (*e.g.*, arrival / service rates, capacities) to be *stationary over time*
- Fails in reality: auto driving, mobile networks, DDoS, ...
- $\Rightarrow$ Consider **Adversarial Network Optimization (ANO)**

Limitation 2: Full Prior-Decision Information

- Many existing network optimization works also require network conditions to be *known a-priori*
- Fails in reality: underwater communication, IoT sensors, ...
- $\Rightarrow$ Consider **Bandit Feedback Models**

Challenges and Main Contributions

(Q). Can we maximize utility in adversarial & multi-hop networks only using bandit feedback?

Why is this HARD?

Challenges and Main Contributions

(Q). Can we maximize utility in adversarial & multi-hop networks only using bandit feedback?

Why is this HARD?

- **ANO.** No reliable statistics. Need worst-case guarantees

Challenges and Main Contributions

(Q). Can we maximize utility in adversarial & multi-hop networks only using bandit feedback?

Why is this HARD?

- **ANO.** No reliable statistics. Need worst-case guarantees
- **Bandit feedback.** No pre-decision info. Only see outcome of chosen action – no counterfactuals

Challenges and Main Contributions

(Q). Can we maximize utility in adversarial & multi-hop networks only using bandit feedback?

Why is this HARD?

- **ANO.** No reliable statistics. Need worst-case guarantees
- **Bandit feedback.** No pre-decision info. Only see outcome of chosen action – no counterfactuals
- **Multi-Hop Topology.** Complex queue inter-dependencies

Challenges and Main Contributions

(Q). Can we maximize utility in adversarial & multi-hop networks only using bandit feedback?

Why is this HARD?

- **ANO.** No reliable statistics. Need worst-case guarantees
- **Bandit feedback.** No pre-decision info. Only see outcome of chosen action – no counterfactuals
- **Multi-Hop Topology.** Complex queue inter-dependencies
- **Utility Maximization.** Unknown, arbitrary, and time-varying utility

Challenges and Main Contributions

(Q). Can we maximize utility in adversarial & multi-hop networks only using bandit feedback?

Why is this HARD?

- **ANO.** No reliable statistics. Need worst-case guarantees
- **Bandit feedback.** No pre-decision info. Only see outcome of chosen action – no counterfactuals
- **Multi-Hop Topology.** Complex queue inter-dependencies
- **Utility Maximization.** Unknown, arbitrary, and time-varying utility

Our Contribution: $\texttt{UMO}^2$

(Utility Max via OLO & BCO)

- **First algorithm** to answer (Q), with rigorous utility & stability guarantees
- **Roadmap:** Online Learning for ANO
- **Novel** OL algs of independent interest

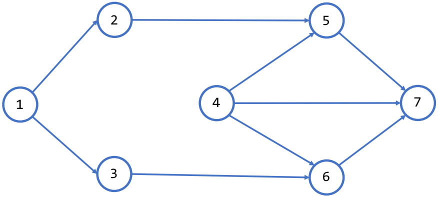
Challenges and Main Contributions (Cont'd)

(Q). Can we maximize utility in adversarial & multi-hop networks only using bandit feedback?

	Network Conditions	Arrival & Service	Topology	Objective	Utility
[Neely et al., 2005]	Stochastic	Known	Multi-Hop	Utility Maximization	Known
[Neely, 2010]	Adversarial	Known	Multi-Hop	Utility Maximization	Known
[Liang and Modiano, 2018b]	Adversarial	Known	Multi-Hop	Network Stability	—
[Liang and Modiano, 2018a]	Adversarial	Known	Multi-Hop	Utility Maximization	Known
[Yang et al., 2023]	Adversarial	Unknown	Single-Hop	Network Stability	—
[Huang et al., 2024]	Adversarial	Unknown	Single-Hop	Network Stability	—
Ours	Adversarial	Unknown	Multi-Hop	Utility Maximization	Unknown

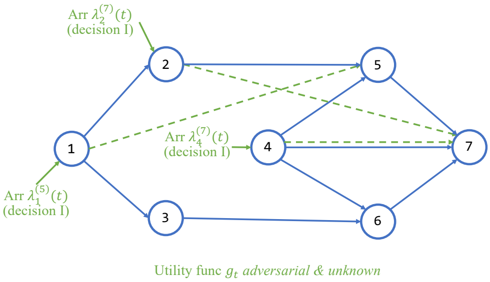
Table: Comparison with Most Related Works

Our Setup: ANO with Bandit Feedback



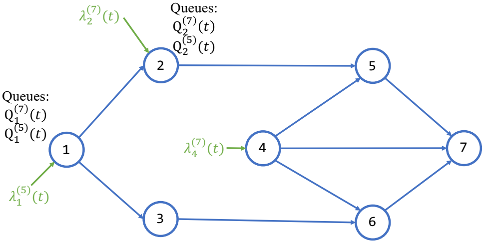
- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning

Our Setup: ANO with Bandit Feedback



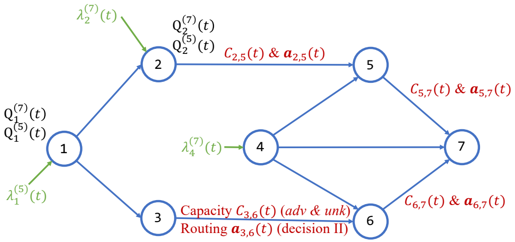
- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning
- **Arrival Decision** $\boldsymbol{\lambda}(t)$ under *adv & unknown* utility func $g_t(\boldsymbol{\lambda}(t))$

Our Setup: ANO with Bandit Feedback



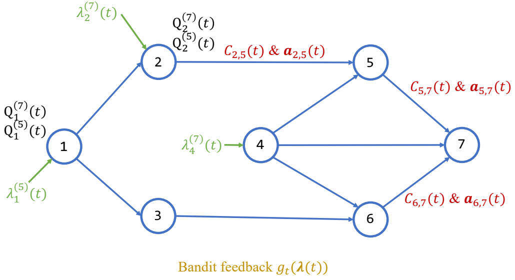
- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning
- **Arrival Decision** $\boldsymbol{\lambda}(t)$ under *adv & unknown* utility func $g_t(\boldsymbol{\lambda}(t))$
- **Multiple Queues** $Q_n^{(k)}(t)$ (#jobs at server n & with destination k)

Our Setup: ANO with Bandit Feedback



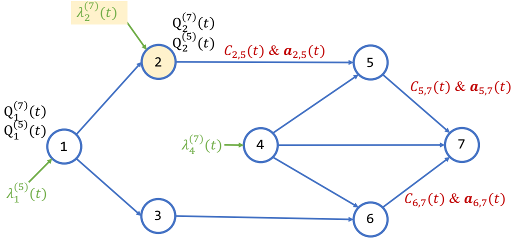
- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning
- **Arrival Decision** $\boldsymbol{\lambda}(t)$ under *adv & unknown* utility func $g_t(\boldsymbol{\lambda}(t))$
- **Multiple Queues** $Q_n^{(k)}(t)$ (#jobs at server n & with destination k)
- **Routing Decision** $\mathbf{a}(t)$ under *adv & unknown* capacity $C_{n,m}(t)$

Our Setup: ANO with Bandit Feedback



- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning
- **Arrival Decision** $\boldsymbol{\lambda}(t)$ under *adv & unknown* utility func $g_t(\boldsymbol{\lambda}(t))$
- **Multiple Queues** $Q_n^{(k)}(t)$ (#jobs at server n & with destination k)
- **Routing Decision** $\mathbf{a}(t)$ under *adv & unknown* capacity $C_{n,m}(t)$
- **Bandit Feedback** after decision: only observes $g_t(\boldsymbol{\lambda}(t))$ but not g_t

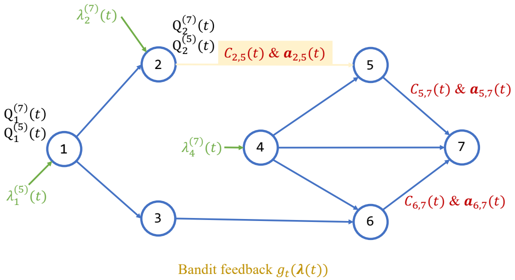
Our Setup: ANO with Bandit Feedback



Bandit feedback $g_t(\boldsymbol{\lambda}(t))$

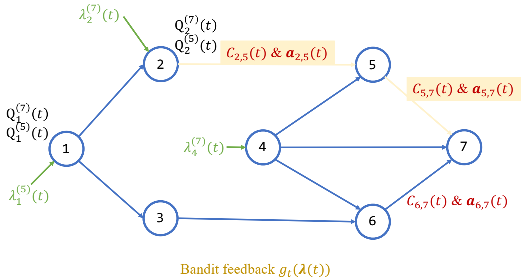
- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning
- **Arrival Decision** $\boldsymbol{\lambda}(t)$ under *adv & unknown* utility func $g_t(\boldsymbol{\lambda}(t))$
- **Multiple Queues** $Q_n^{(k)}(t)$ (#jobs at server n & with destination k)
- **Routing Decision** $\boldsymbol{a}(t)$ under *adv & unknown* capacity $C_{n,m}(t)$
- **Bandit Feedback** after decision: only observes $g_t(\boldsymbol{\lambda}(t))$ but not g_t

Our Setup: ANO with Bandit Feedback



- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning
- **Arrival Decision** $\boldsymbol{\lambda}(t)$ under *adv & unknown* utility func $g_t(\boldsymbol{\lambda}(t))$
- **Multiple Queues** $Q_n^{(k)}(t)$ (#jobs at server n & with destination k)
- **Routing Decision** $\mathbf{a}(t)$ under *adv & unknown* capacity $C_{n,m}(t)$
- **Bandit Feedback** after decision: only observes $g_t(\boldsymbol{\lambda}(t))$ but not g_t

Our Setup: ANO with Bandit Feedback



- **Multi-Hop Network** $\mathcal{G} = (\mathcal{N}, \mathcal{L})$; T-round planning
- **Arrival Decision** $\boldsymbol{\lambda}(t)$ under *adv & unknown* utility func $g_t(\boldsymbol{\lambda}(t))$
- **Multiple Queues** $Q_n^{(k)}(t)$ (#jobs at server n & with destination k)
- **Routing Decision** $\mathbf{a}(t)$ under *adv & unknown* capacity $C_{n,m}(t)$
- **Bandit Feedback** after decision: only observes $g_t(\boldsymbol{\lambda}(t))$ but not g_t

Online Learning for ANO: 3-Step Punchline

1. **Reduce** to Online Learning via *global Lyapunov analysis*
   - single Lyapunov term is too sensitive to adversarial corruptions

Online Learning for ANO: 3-Step Punchline

1. **Reduce** to Online Learning via *global Lyapunov analysis*
   - single Lyapunov term is too sensitive to adversarial corruptions

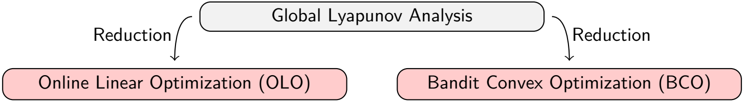
Global Lyapunov Analysis

Reduction → Online Linear Optimization (OLO)

Reduction → Bandit Convex Optimization (BCO)

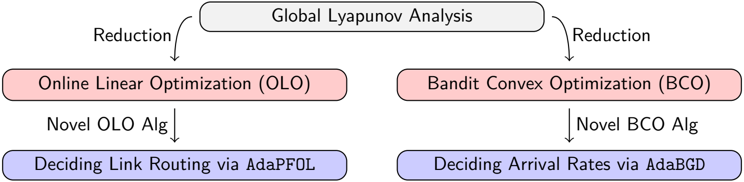
Online Learning for ANO: 3-Step Punchline

1. **Reduce** to Online Learning via *global Lyapunov analysis*
   - single Lyapunov term is too sensitive to adversarial corruptions
2. **Design** novel *queue-based OL* for ANO-specific challenges
   - unbounded queue lengths $\Rightarrow$ unbounded loss magnitudes



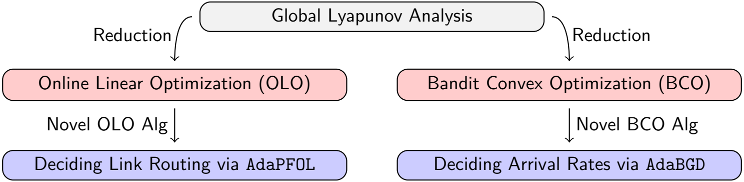
Online Learning for ANO: 3-Step Punchline

1. **Reduce** to Online Learning via *global Lyapunov analysis*
   - single Lyapunov term is too sensitive to adversarial corruptions
2. **Design** novel *queue-based OL* for ANO-specific challenges
   - unbounded queue lengths ⇒ unbounded loss magnitudes



Online Learning for ANO: 3-Step Punchline

1. **Reduce** to Online Learning via *global Lyapunov analysis*
   - single Lyapunov term is too sensitive to adversarial corruptions
2. **Design** novel *queue-based OL* for ANO-specific challenges
   - unbounded queue lengths $\Rightarrow$ unbounded loss magnitudes
3. **Extend** OL guarantees to ANO via *self-bounding analysis*
   - $\mathbb{E}[\sum_t \|\boldsymbol{Q}(t)\|_1] \le T^{1/4}\,\mathbb{E}[\sqrt{\sum_t \|\boldsymbol{Q}(t)\|_2^2}] \Rightarrow$ network stable



Online Learning for ANO: 3-Step Punchline

1. **Reduce** to Online Learning via *global Lyapunov analysis*
   - single Lyapunov term is too sensitive to adversarial corruptions
2. **Design** novel *queue-based OL* for ANO-specific challenges
   - unbounded queue lengths $\Rightarrow$ unbounded loss magnitudes
3. **Extend** OL guarantees to ANO via *self-bounding analysis*
   - $\mathbb{E}[\sum_t \|\boldsymbol{Q}(t)\|_1] \le T^{1/4}\,\mathbb{E}[\sqrt{\sum_t \|\boldsymbol{Q}(t)\|_2^2}] \Rightarrow$ network stable

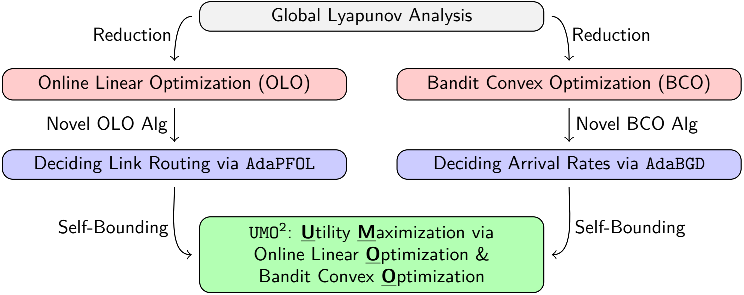


Figure: Analysis Roadmap of Our Algorithm UMO^2

Algorithms & Analysis Outlines

In each round
$t = 1, 2, \ldots, T$

Figure: `UMO`^{2} for Utility Maximization

Algorithms & Analysis Outlines

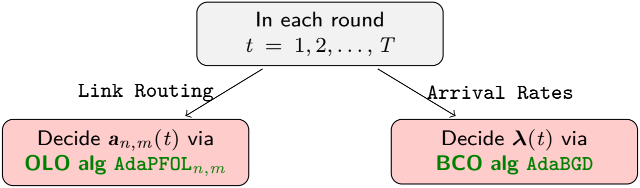


Figure: `UMO`^{2} for Utility Maximization

Algorithms & Analysis Outlines

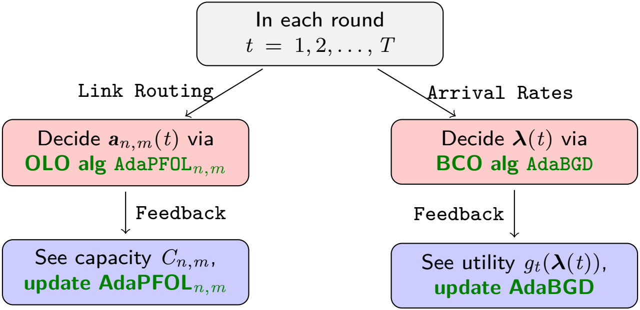


Figure: $\texttt{UMO}^2$ for Utility Maximization

Algorithms & Analysis Outlines

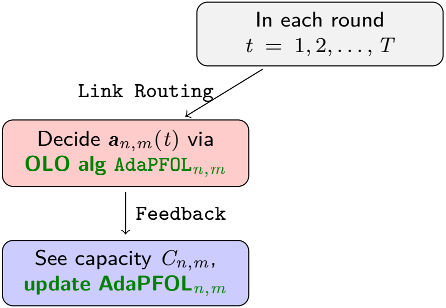


Figure: `NSO` for Network Stability

Network Stability $\Rightarrow$ Online Linear Optimization

Step 1: Reduction via Global Lyapunov Analysis

Stability $\min \mathbb{E}\left[\sum_t \|\boldsymbol{Q}(t)\|_1\right]$

Network Stability ⇒ Online Linear Optimization

Step 1: Reduction via Global Lyapunov Analysis

$$\text{Stability} \quad \min \mathbb{E}\left[\sum\nolimits_t \|\boldsymbol{Q}(t)\|_1\right]$$

$$\Downarrow \quad \text{(Global Lyapunov drift analysis)}$$

$$\forall (n,m) \in \mathcal{L}, \min \mathbb{E}\left[\sum\nolimits_t \langle \boldsymbol{a}_{n,m}(t)\,,\ C_{n,m}(t)(\boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t))\,\rangle\right]$$

Network Stability $\Rightarrow$ Online Linear Optimization

Step 1: Reduction via Global Lyapunov Analysis

Stability $\min \mathbb{E}\left[\sum_t \|\boldsymbol{Q}(t)\|_1\right]$

$\Downarrow$ (Global Lyapunov drift analysis)

$$\forall (n,m) \in \mathcal{L}, \min \mathbb{E}\left[\sum_t \langle \underbrace{\mathbf{a}_{n,m}(t)}_{\text{Decision}}, C_{n,m}(t)(\boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t)) \rangle\right]$$

Network Stability $\Rightarrow$ Online Linear Optimization

Step 1: Reduction via Global Lyapunov Analysis

Stability $\min \mathbb{E}\left[\sum_t \|\boldsymbol{Q}(t)\|_1\right]$

$\Downarrow$ (Global Lyapunov drift analysis)

$$\forall (n,m) \in \mathcal{L},\ \min \mathbb{E}\left[\sum_t \langle \underbrace{\mathbf{a}_{n,m}(t)}_{\text{Decision}}, \underbrace{C_{n,m}(t)(\boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t))}_{\text{Loss Vector}} \rangle\right]$$

$\Longrightarrow$ **Reduce to Online Linear Optimization (OLO)**

Network Stability $\Rightarrow$ Online Linear Optimization

Step 1: Reduction via Global Lyapunov Analysis

$$\text{Stability} \quad \min \mathbb{E}\left[\sum\nolimits_t \|\boldsymbol{Q}(t)\|_1\right]$$

$$\Downarrow \quad \text{(Global Lyapunov drift analysis)}$$

$$\forall (n,m) \in \mathcal{L}, \min \mathbb{E}\left[\sum\nolimits_t \langle \underbrace{\mathbf{a}_{n,m}(t)}_{\text{Decision}}, \underbrace{C_{n,m}(t)(\boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t))}_{\text{Loss Vector}} \rangle\right]$$

$\Longrightarrow$ **Reduce to Online Linear Optimization (OLO)**

ANO-Specific Challenges

1. **Unbounded Loss Magnitudes**

Network Stability $\Rightarrow$ Online Linear Optimization

Step 1: Reduction via Global Lyapunov Analysis

$$\text{Stability} \quad \min \mathbb{E}\left[\sum_t \|\boldsymbol{Q}(t)\|_1\right]$$

$$\Downarrow \quad \text{(Global Lyapunov drift analysis)}$$

$$\forall (n,m) \in \mathcal{L}, \min \mathbb{E}\left[\sum_t \langle \underbrace{\mathbf{a}_{n,m}(t)}_{\text{Decision}}, \underbrace{C_{n,m}(t)(\boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t))}_{\text{Loss Vector}} \rangle\right]$$

$\Longrightarrow$ **Reduce to Online Linear Optimization (OLO)**

ANO-Specific Challenges

1. **Unbounded Loss Magnitudes**
2. **Negative Loss Components**

Network Stability ⇒ Online Linear Optimization

Step 1: Reduction via Global Lyapunov Analysis

Stability $\min \mathbb{E}\left[\sum_t \|\boldsymbol{Q}(t)\|_1\right]$

$\Downarrow$ (Global Lyapunov drift analysis)

$$\forall(n,m)\in\mathcal{L},\ \min \mathbb{E}\left[\sum_t \langle \underbrace{\mathbf{a}_{n,m}(t)}_{\text{Decision}}, \underbrace{C_{n,m}(t)(\boldsymbol{Q}_m(t)-\boldsymbol{Q}_n(t))}_{\text{Loss Vector}} \rangle\right]$$

$\Longrightarrow$ **Reduce to Online Linear Optimization (OLO)**

ANO-Specific Challenges

1. **Unbounded Loss Magnitudes**
2. **Negative Loss Components**
3. **Requires Adaptivity**

Existing OLO algorithms fail to tackle all issues simultaneously.

OLO Guarantee ⇒ Network Stability Guarantee

Step 2: A Novel & Customized OLO algorithm: `AdaPFOL`

$$\mathcal{O}\left(\sqrt{\sum_t \| C_{n,m}(t)(\boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t))\|_2^2}\right)$$

OLO Guarantee ⇒ Network Stability Guarantee

Step 2: A Novel & Customized OLO algorithm: `AdaPFOL`

$$\mathcal{O}\left(\sqrt{\sum_t \lVert C_{n,m}(t)(\boldsymbol{Q}_m(t)-\boldsymbol{Q}_n(t))\rVert_2^2}\right) \lesssim \mathcal{O}\left(\sqrt{\sum_t \lVert \boldsymbol{Q}(t)\rVert_2^2}\right)$$

allows **large magnitudes**, **negative losses**, and is **adaptive**

OLO Guarantee ⇒ Network Stability Guarantee

Step 2: A Novel & Customized OLO algorithm: `AdaPFOL`

$$\mathcal{O}\left(\sqrt{\sum_t \| C_{n,m}(t)(\boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t))\|_2^2}\right) \lesssim \mathcal{O}\left(\sqrt{\sum_t \|\boldsymbol{Q}(t)\|_2^2}\right)$$

allows **large magnitudes**, **negative losses**, and is **adaptive**

Step 3: Self-Bounding Analysis for ANO Guarantee

Self-bounding analysis: From Steps 1 & 2, `NSO` ensures

$$\mathbb{E}\left[\sum_t \|\boldsymbol{Q}(t)\|_1\right] \le \mathcal{O}(T^{1/4})\,\mathbb{E}\left[\sqrt{\sum_t \|\boldsymbol{Q}(t)\|_2^2}\right] \le \mathcal{O}(T^{1/4})\,\mathbb{E}\left[\sum_t \|\boldsymbol{Q}(t)\|_1\right]^{3/4},$$

$\Longrightarrow \frac{1}{T}\,\mathbb{E}[\sum_t \|\boldsymbol{Q}(t)\|_1] = \mathcal{O}(1)$, *i.e.*, network is stabilized by `NSO`

Network Stability to Utility Maximization: Another 3 Steps

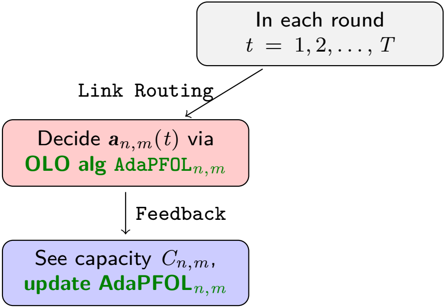


Figure: `NSO` for Network Stability

Network Stability to Utility Maximization: Another 3 Steps

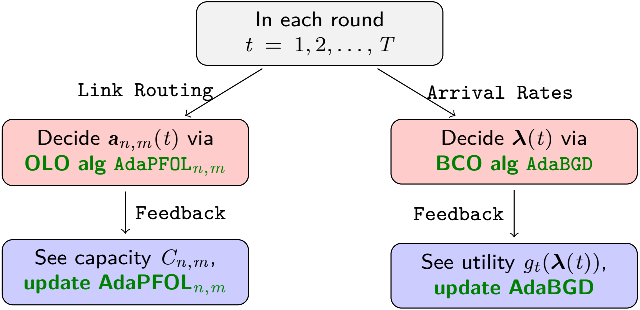


Figure: UMO^2 for Utility Maximization

Analysis Sketch for UMO^2

Step 1: Global Lyapunov Drift-plus-Penalty (DPP) Analysis

$$\Longrightarrow \min \underbrace{\mathbb{E}[\textstyle\sum_t \sum_{(n,m)\in\mathcal{L}} C_{n,m}(t)\langle \boldsymbol{a}_{n,m}(t), \boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t)\rangle]}_{\text{OLO Regret (handled by AdaPFOL)}}$$

$$+ \quad \underbrace{\mathbb{E}[\textstyle\sum_t \langle \boldsymbol{Q}(t), \boldsymbol{\lambda}(t)\rangle - V g_t(\boldsymbol{\lambda}(t))]}_{\text{Bandit Convex Optimization (BCO) Regret}} \qquad (V = o_T(1) \text{ is param})$$

Analysis Sketch for UMO^2

Step 1: Global Lyapunov Drift-plus-Penalty (DPP) Analysis

$$\Longrightarrow \min \underbrace{\mathbb{E}[\textstyle\sum_t \sum_{(n,m)\in\mathcal{L}} C_{n,m}(t)\langle \boldsymbol{a}_{n,m}(t), \boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t)\rangle]}_{\text{OLO Regret (handled by AdaPFOL)}}$$

$$+ \quad \underbrace{\mathbb{E}[\textstyle\sum_t \langle \boldsymbol{Q}(t), \boldsymbol{\lambda}(t)\rangle - V g_t(\boldsymbol{\lambda}(t))]}_{\text{Bandit Convex Optimization (BCO) Regret}} \qquad (V = o_T(1) \text{ is param})$$

Step 2-1: AdaPFOL

$$\text{Reg} \lesssim \mathcal{O}\left(\sqrt{\textstyle\sum_t \|\boldsymbol{Q}(t)\|_2^2}\right)$$

Analysis Sketch for UMO^2

Step 1: Global Lyapunov Drift-plus-Penalty (DPP) Analysis

$$\Longrightarrow \min \underbrace{\mathbb{E}[\textstyle\sum_t \sum_{(n,m)\in\mathcal{L}} C_{n,m}(t)\langle \boldsymbol{a}_{n,m}(t), \boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t)\rangle]}_{\text{OLO Regret (handled by AdaPFOL)}}$$

$$+ \underbrace{\mathbb{E}[\textstyle\sum_t \langle \boldsymbol{Q}(t), \boldsymbol{\lambda}(t)\rangle - V g_t(\boldsymbol{\lambda}(t))]}_{\text{Bandit Convex Optimization (BCO) Regret}} \qquad (V = o_T(1) \text{ is param})$$

Step 2-1: AdaPFOL

$$\text{Reg} \lesssim \mathcal{O}\left(\sqrt{\textstyle\sum_t \|\boldsymbol{Q}(t)\|_2^2}\right)$$

Step 2-2: AdaBGD

$$\text{Reg} \lesssim \mathcal{O}\left(\left(\textstyle\sum_t \|\boldsymbol{Q}(t)\|_2^{4/3}\right)^{3/4}\right)$$

Analysis Sketch for UMO^2

Step 1: Global Lyapunov Drift-plus-Penalty (DPP) Analysis

$$\Longrightarrow \min \underbrace{\mathbb{E}[\textstyle\sum_t \sum_{(n,m)\in\mathcal{L}} C_{n,m}(t)\langle \boldsymbol{a}_{n,m}(t), \boldsymbol{Q}_m(t) - \boldsymbol{Q}_n(t)\rangle]}_{\text{OLO Regret (handled by AdaPFOL)}}$$

$$+ \quad \underbrace{\mathbb{E}[\textstyle\sum_t \langle \boldsymbol{Q}(t), \boldsymbol{\lambda}(t)\rangle - V g_t(\boldsymbol{\lambda}(t))]}_{\text{Bandit Convex Optimization (BCO) Regret}} \qquad (V = o_T(1) \text{ is param})$$

Step 2-1: AdaPFOL

$$\text{Reg} \lesssim \mathcal{O}\left(\sqrt{\textstyle\sum_t \|\boldsymbol{Q}(t)\|_2^2}\right)$$

Step 2-2: AdaBGD

$$\text{Reg} \lesssim \mathcal{O}\left(\left(\textstyle\sum_t \|\boldsymbol{Q}(t)\|_2^{4/3}\right)^{3/4}\right)$$

Step 3: More Sophisticated Self-Bounding Analysis

1. *Coarse queue length.* 2-1&2 $\overset{\text{Self Bounding}}{\Longrightarrow}$ $\mathbb{E}[\sum_t \|\boldsymbol{Q}(t)\|_1] = \mathcal{O}(VT)$
2. *Utility gap.* Queue length $\mathcal{O}(VT) \Longrightarrow$ utility gap $= \mathcal{O}(V^{-1})$
3. *Refined queue length.* 1+2 $\overset{\text{Self Bounding}}{\Longrightarrow}$ $\mathbb{E}[\sum_t \|\boldsymbol{Q}(t)\|_1] = \mathcal{O}(T)$

Final Guarantee of UMO^2

UMO^2 Ensures Simultaneously...

- **Network Stability.** Average queue length remain bounded:

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\|\boldsymbol{Q}(t)\|_1\right]=\mathcal{O}_T(1)$$

Final Guarantee of UMO^2

UMO^2 Ensures Simultaneously...

- **Network Stability.** Average queue length remain bounded:

$$\frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T}\|\boldsymbol{Q}(t)\|_1\right] = \mathcal{O}_T(1)$$

- **Near-Optimal Utility.** Utility converges to the best "mildly varying" policy with $1/\mathrm{poly}(T)$ gap (see our Thm 4.5):

$$\frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T} g_t(\mathring{\boldsymbol{\lambda}}(t)) - g_t(\boldsymbol{\lambda}(t))\right] = \mathcal{O}_T(1/\mathrm{poly}(T))$$

Final Guarantee of UMO^2

UMO^2 Ensures Simultaneously...

- **Network Stability.** Average queue length remain bounded:

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\|\boldsymbol{Q}(t)\|_1\right]=\mathcal{O}_T(1)$$

- **Near-Optimal Utility.** Utility converges to the best "mildly varying" policy[a] with $1/\text{poly}(T)$ gap (see our Thm 4.5):

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}g_t(\mathring{\boldsymbol{\lambda}}(t))-g_t(\boldsymbol{\lambda}(t))\right]=\mathcal{O}_T(1/\text{poly}(T))$$

[a] Any forward-looking (*i.e.*, allowing offline/hindsight optimization) policy with slowly changing actions.

Main Results & Takeaway

Main Results

- **First** utility result for multi-hop ANO w/ bandit feedback
- **General** reduction framework from ANO to Online Learning
- **Novel** adaptive OL algs (`AdaPFOL`, `AdaBGD`) for unique network challenges (*esp.* unbounded queue & self-bounding)

Online Learning for ANO: 3-Step Punchline

1. **Reduce** to Online Learning via Global Lyapunov analyses
2. **Design** novel & customized OL algs for ANO challenges
3. **Extend** OL guarantee to ANO via self-bounding analysis

Questions are more than welcomed!

References I

Jim G Dai and Wuqin Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53 (2):197–218, 2005.

Atilla Eryilmaz and Rayadurgam Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM transactions on networking*, 15(6):1333–1344, 2007.

Leonidas Georgiadis, Michael J Neely, Leandros Tassiulas, et al. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends® in Networking*, 1(1):1–144, 2006.

Jiatai Huang, Leana Golubchik, and Longbo Huang. When lyapunov drift based queue scheduling meets adversarial bandit learning. *IEEE/ACM Transactions on Networking*, 2024.

Libin Jiang and Jean Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 18(3):960–972, 2009.

Qingkai Liang and Evtan Modiano. Network utility maximization in adversarial environments. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 594–602. IEEE, 2018a.

Qingkai Liang and Eytan Modiano. Minimizing queue length regret under adversarial network models. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):1–32, 2018b.

Xiaojun Lin and Ness B Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM transactions on networking*, 14(2):302–315, 2006.

Siva Theja Maguluri, Rayadurgam Srikant, and Lei Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. In *2012 Proceedings IEEE Infocom*, pages 702–710. IEEE, 2012.

Xiaoqiao Meng, Vasileios Pappas, and Li Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010.

Michael J Neely. Universal scheduling for networks with arbitrary traffic, channels, and mobility. In *49th IEEE Conference on Decision and Control (CDC)*, pages 1822–1829. IEEE, 2010.

MJ Neely, E Modiano, and Chih-Ping Li. Fairness and optimal stochastic control for heterogeneous networks. 3: 1723–1734, 2005.

References II

Mohammad Rahdar, Lizhi Wang, and Guiping Hu. A tri-level optimization model for inventory control with uncertain demand and lead time. *International Journal of Production Economics*, 195:96–105, 2018.

Devavrat Shah and Jinwoo Shin. Randomized scheduling algorithm for queueing networks. *Annals of Applied Probability*, 22(1):128–171, 2012.

Rayadurgam Srikant and Lei Ying. *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37 (12):1936–1948, 1992.

Leandros Tassiulas and Anthony Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information theory*, 39(2):466–478, 2002.

Zixian Yang, R Srikant, and Lei Ying. Learning while scheduling in multi-server systems with unknown statistics: Maxweight with discounted ucb. In *International Conference on Artificial Intelligence and Statistics*, pages 4275–4312. PMLR, 2023.