Robust Resource Provisioning in Time-Varying Edge Networks

Ruozhou Yu, North Carolina State University
Guoliang Xue, Yinxin Wan, Arizona State University
Jian Tang, Syracuse University
Dejun Yang, Colorado School of Mines
Yusheng Ji, National Institute of Informatics, Japan
Geo-Distributed Services & Edge Computing
Time-Varying Demands in Geo-Distributed Apps
Patterns in Real-world Datasets

**Dartmouth College Wireless APs**
- Top 10 APs with highest loads
- Load: avg. # devices / hour
- Averaged over a year (9/2002-9/2003)

**NYC Yellow Taxi 2018**
- Top 10 zones w/ most drop-offs
- Load: avg. # passenger drop-offs
- Averaged over a year

**Observation 1:** Non-i.i.d. demand distributions across time & locations.

**Observation 2:** Repeating / seasonal patterns in temporal domain.
Resource Provisioning for Edge Services

- **Inputs:** edge network (edge nodes), app/service, demands
- **Outputs:** 1) app/service hosting, 2) traffic routing / engineering
- Studied in the literature, e.g. [1][2], …
- … but with **static inputs**!

---


Methodology Overview

Inputs:
- **App Demands**: Time-varying, geo-distributed
- **Edge Network**: Topology, time-varying delays

System-wide Optimization:
- **Abstract System Model**:
  - Time-varying model
  - System risk model (CVaR)
  - Three-stage stochastic optimiz.

Optimization Framework:
- Nested Bender decomposition
- Efficient subproblem solving

Outputs:
- **Service Deployment**: Global fixed decisions
- **Net Provisioning**: Per-time slot decisions
- **Net Estimation**: Per-realization decisions
System Model: Involved Parties

Edge Service Provider
- Submits service requests
- Measures and predicts demands
  - Dynamically balances load

Edge Computing Manager(s)
- Manages edge nodes & resources
  - Decides computing costs

Network Manager
- Manages edge network
  - Decides network policies
  - Provisions network resources (bw)
Edge Network: A General Model

- **Challenge:** heterogeneous network environments

- **Model:** general directed graph $G=(N, L)$, with edge nodes $H$ and APs $A$
  - Weights: link bandwidth, $<\text{link delay}>$, edge node cost, $<\text{AP demand}>$

**Wireless RANs:**
- Geo-distributed
- Limited capacity
- Interference

**Edge Network:**
- Complex topo
- Distributed
- Dynamic load

**Backbones:**
- Large-scale
- High latency
- ISP policies
Edge Demand Model

- **Challenge:** non-static, time-varying
- **Observation:** seasonal/repeating patterns
  - *Example:* the load in the same hour of workdays at an AP is similar

**Repeating time-slotted demand model**

- Demand across slots in one period: non-i.i.d.
- Demand per slot across periods: i.i.d.
Edge Resource Provisioning /1

- **Challenges:** which decisions should be dynamic, which static?
- **Formulation:** a three-stage decision problem

- **Stage 1: Service Deployment (SD)**
  - Deploy edge service on host nodes by ECM
  - Globally fixed: static across time slots & periods.

- **Stage 2: Network Provisioning (NPR)**
  - Network routing and bandwidth allocation by NM
  - Per-slot: dynamic across time slots, but static for same slot across periods!

- **Stage 3: Network Estimation (NE)**
  - Instantaneous traffic allocation by ESP
  - Dynamic: dynamic across both time slots and periods!
Objective and Overall Formulation

Objective: minimize max traffic-averaged delay across time slots

\[
\begin{align*}
\min_{\chi \in \mathcal{F}} \max_t \{D_t\}, \\
\sum_{h \in H} c_h x(h) &\leq C. \\
\sum_{h \in H} x(h) &\geq 1.
\end{align*}
\]

\(s.t.
\)

\[
\begin{align*}
y(t, p) &\leq b_p \max_x x(h), \quad \forall t, p \in P, \\
\sum_{p \in P : l \in p} y(t, p) &\leq b_l, \quad \forall t, l \in L.
\end{align*}
\]

Stage 1: SD

Stage 2: NPR

Stage 3: NE

\[
\begin{align*}
z(t, p) &\leq y(t, p), \quad \forall t, p \in P. \\
\sum_{p \in P_a} z(t, p) &\geq \delta_{t, a}, \quad \forall t, a \in A,
\end{align*}
\]

But \(\{\delta_{t, a}\}\) and \(\{d_{t, p}\}\) are both random…
SO and CVaR

- **Stochastic Optimization (SO):** optimize a function in presence of randomness (random objective and/or constraints)
  - Traditional approach: expectation optimization
    \[
    \min_{x \in \mathcal{F}} \max_t \mathbb{E}[D_t]
    \]
  - **Issue:** unbounded risk in rare but unfortunate scenarios
    - E.g., abnormal demands due to public events, rare large-scale failures, …

- How to model these *unfortunate scenarios*?
- **Value-at-Risk (VaR) and Conditional-Value-at-Risk (CVaR):**
  - Widely used in economics and finance
  - \( \text{VaR}_\alpha(R) = \min \{ c \in \mathbb{R} \mid R \text{ does not exceed } c \text{ with at least } \alpha \text{ prob.} \} \)
  - \( \text{CVaR}_\alpha(R) = \mathbb{E}[R \mid R \geq \text{VaR}_\alpha(R)] \)
    - Expectation of \( R \) in the worst \((1-\alpha)\) scenarios
- **Our approach:** optimize both expectation and CVaR
  \[
  \min_{x \in \mathcal{F}} \max_t \{ \rho_1 \cdot \mathbb{E}[D_t] + \rho_2 \cdot \text{CVaR}_\alpha(D_t) \}, \quad (11)
  \]
The Robust Edge Provisioning (REP) problem

\[
\max_t \quad \text{Linearization}
\]

\[
\min \quad X, Y, Z, R, W, D
\]

\[
D \geq \rho_1 \frac{1}{K} \sum_{k=1}^{K} D_t^k +
\]

\[
\rho_2 \left( r(t) + \frac{1}{1-\alpha} \frac{1}{K} \sum_{k=1}^{K} w(t, k) \right), \forall t;
\]

\[
w(t, k) \geq D_t^k - r(t), \forall t, k;
\]

(1)-(4), and \(\forall t, k, (5)-(6)\).

\[
\text{CVaR LP Transformation (Rockafella & Uryasev)}
\]

MILP with \(\Theta(TKP)\) variables.

NP-hard by reduction from Knapsack.
Outlines

- Background and Motivation
- System Modeling
- Algorithm Design and Analysis
- Performance Evaluation
- Discussions, Future Work and Conclusions
Iterative Optimization Algorithm

- **Benders’ decomposition**: (Row Generation) In each iteration, add new constraints (cuts) to the problem that push the main problem towards the optimal:
  - INIT: feasible main solution; then proceed in iterations:
    - Solve sub dual problem based on main solution (UB).
    - If sub dual unbounded, add feasibility cut to main; if sub dual optimal, add optimality cut to main.
    - Solve updated main (LB).
  - Until UB – LB < ε.

- **Nested Benders’ decomposition**
  - Apply two Benders’ decompositions for Phase-I and Phase-II respectively.

Convergence to **optimality**: proof by Benders.
Additional Techniques Applied

- **Multiple Cuts** (Birge & Louveaux)
  - Dividing one optimality cut into one cut per sub-problem.
  - Improves efficiency by pruning more sub-optimal region per-iteration.

- **Fast Forward Fast Backward (FFFB)**
  - Do not wait till Phase-II convergence to update Phase-I main problem.
  - Cuts based on non-optimal Phase-II solutions help prune more sub-optimal region per-iteration.

- **Analytical Stage-3 Dual Solving**
  - Linear time algorithm for solving the Stage-3 dual problems…
  - … instead of cubic time for solving as an LP
Simulation Settings

- **Settings**
  - **Dataset:** NYC Yellow Taxi 2018
    - 12 months of Taxi drop-off data (~112 million taxi trips)
    - Picked 5 or 20 most popular zones out of 262 (18% or 55% of all demands)
    - 100-days for training: solving SAA formulation for SD and NPR
    - 265-days for testing: evaluating solutions with NE

- **Synthetic Data**
  - Random topologies: Watts-Strogatz with $k = 4$ and $p = 0.3$ (5 edge nodes)
  - Deployment costs: $\mathcal{N}(1000, 200^2)$; cost budget: 3300 (uniform)
  - Pathbook: 3 min-hop paths for each AP-Edge node pair
  - Network conditions:
    - Normal scenario: 5 Gbps links with $\mathcal{N}(10, 4^2)$ ms delays
    - Congested scenario: 2 Gbps links with half nodes experiencing $50 \times$ delays

- $\rho_1 = \rho_2 = 0.5$ (expectation vs. CVaR), $\alpha = 0.95$ (CVaR confidence), $\varepsilon = 10^{-3}$ (convergence)
Experiment Results

**Time-varying vs. Time-agnostic**

- Time-varying has increased advantage over time-agnostic with more slots.
  => **Fixed provisioning** without per-slot adjustment has poor performance.
  (For each slot, load is averaged over entire slot.)

**Setting**: Small/Congested
Experiment Results

(a) Training objective value  
(b) Testing objective value

Optimal vs. Heuristics
• Consistent performance advantage over heuristics  
  => User satisfaction / revenue in the long-term

RAND: random edge node  
AVG: optimiz. avg. delay  
Setting: Medium/Normal
Outlines

- Background and Motivation
- System Modeling
- Algorithm Design and Analysis
- Performance Evaluation
- Discussions, Future Work and Conclusions
Other Perspectives, Conclusions

So far, we’ve talked about
- Model: time-varying demands & network
- CVaR w/ multi-stage stochastic optimization
- Provisioning with single service & pathbook

What could be improved
- Multi-service provisioning / sharing
- Dynamic routing w/o pathbook
- Multi-dimensional network resources
- Distribution-aware formulations
- Improved optimization methods
- Learning-based optimization

Conclusions: observed uncertainties => risk-aware networking
Thank you very much!

Q&A?