Application of Optimization Methods and Edge AI

Hai-Liang Zhao
hliangzhao97@gmail.com

November 17, 2018

This slide can be downloaded at ▸ *Link*.

Outline

1. Application of Optimization Methods
   - ▷ Existing Methods and Their Applications
   - ▷ How to Design Novel Models with Methods Embedded Naturally?

Outline

1. Application of Optimization Methods
   - ▷ Existing Methods and Their Applications
   - ▷ How to Design Novel Models with Methods Embedded Naturally?

2. Edge AI
   - ▷ Existing Paradigms
   - ▷ Preparation for Designing `Egent`

Outline

Regular Optimization Methods

1. Evolutionary Algorithms
2. Lyapunov Optimization
3. Stochastic Programming
4. Game Theory
5. Traditional Machine Learning Methods
6. Various Programming Methods
7. Deep Reinforcement Learning
8. ...

Evolutionary Algorithms

1. Swarm Intelligence
2. Tabu Search
3. Simulated Annealing
4. Artificial Neural Networks
5. Population-based Algorithms
    1. genetic algorithm
    2. particle swarm optimization
    3. negative selection algorithm
    4. learning-teaching-based optimization
6. Too many of them ...

Many works on ***Service Composition*** contributed by Prof. Shuiguang Deng are solved by ***Evolutionary Algorithms*** because they are **method-free**.

Lyapunov Optimization

Standard Lyapunov Optimization is a trump card for *stochastic optimization problems*.

1. Virtual Queues
2. Drift-Plus-Penalty Expression
3. Approximate Scheduling
4. Performance Analysis
    1. average penalty analysis
    2. average queue size analysis
5. Trade-off by Tuning V

A brief introdution for researchers can be found at ▸ *Link*.

Applications of Lyapunov Optimization

Yuyi Mao's papers are inundated with this kind of methods.

1. Match with Lyapunov Optimization Methods
   1. Construct Virtual Queues for Constraints
   2. Replace the Original Problem with a Deterministic one
   3. Solve the Approximate-Convex Problem with **Ingenious Mathematic Tricks**
2. Utilize Lagrange Methods and KKT Conditions
3. Performence Analysis: $O(V), O(\frac{1}{V})$

Apperently Yuyi Mao acquires prociency in Michael. J. Neely's book: ***Stochastic Network Optimization with Application to Communication and Queueing Systems***

Extensions on Lyapunov Optimization

1. Extensions to Variable Frame Length Systems (Dynamic Optimization and Learning for **Renewal Systems**)
2. Combination with **Lagrange Multipliers**
3. Network Utility Maximization over Partially Observable **Markovian Channels**
4. Under **Non-Convex** Problems (Greedy primal-dual algorithm)

My work

$$\mathcal{P}: \max_{\boldsymbol{I}^t} \lim_{T\to+\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i\in\mathcal{N}} \mathcal{U}_i(\boldsymbol{I}^t_{i,:})\right],$$

where $\mathcal{U}_i(\boldsymbol{I}^t_{i,:})$ is defined as

$$\mathcal{U}_i(\boldsymbol{I}^t_{i,:}) \triangleq \sum_{j\in\mathcal{M}} r^t_{i,j} I^t_{i,j} - \phi^t_i \cdot \left[\sum_{j\in\mathcal{M}} \epsilon^t_{i,j} I^t_{i,j} - \psi^{safe}_i, 0\right]^+.$$

$$\begin{aligned} s.t. \qquad & \boldsymbol{I}^t \in \{0,1\}^{N\times M}, t\in\mathcal{T}, \\ & \textstyle\sum_{i\in\mathcal{N}} I^t_{i,j} \le N^{max}_j, j\in\mathcal{M}, t\in\mathcal{T}, \\ & \textstyle\sum_{j\in\mathcal{M}} \epsilon^t_{i,j} I^t_{i,j} \le \psi^t_i, i\in\mathcal{N}, t\in\mathcal{T}. \end{aligned}$$

Stochatsic Programming

Two-stage or ***Multi-stage*** $\Downarrow$

1. Scenario construction
2. Monte Carlo techniques (**SAA method**)
3. Evaluation Candidate Solutions (measure the **optimality gap** between the optimal value and the estimated value)

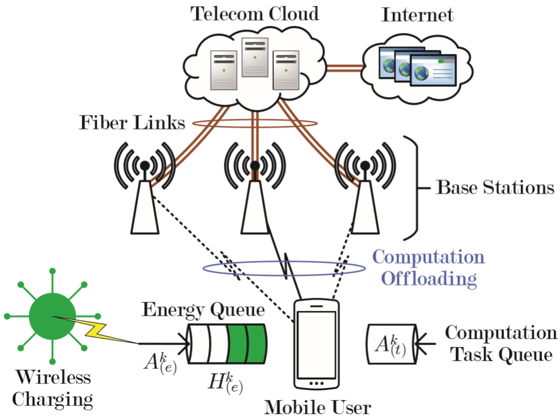
My work

$$
\begin{aligned}
\mathcal{P}: \quad \min_{\boldsymbol{\Theta}' \in \mathcal{Q}} \quad & \mathbb{E}_P[C(\boldsymbol{\Theta}', \mathbf{D})] \triangleq \sum_{i \in \mathcal{N}} \Bigg(I_i \cdot \mathbb{E}_P[c_i^{\text{local}\star}] \\
& + \sum_{j \in \mathcal{M}_k} O_{ij} \cdot \mathbb{E}_P[c_{ij}^{\text{tx}\star}] \\
& + \sum_{j \in \mathcal{M}_k} \sum_{j' \in \mathcal{M}_{k_j} \setminus \{j\}} R_{ij'}^{k_j} \cdot \mathbb{E}_P[c_{ijj'}^{\text{reloc}\star}] \\
& + \sum_{j \in \mathcal{M}_k} \sum_{j' \in \mathcal{M}_k} R_{ij'}^{k_j} \cdot \mathbb{E}_P[c_{ij}^{\text{server}\star}] \Bigg),
\end{aligned}
$$

with 8 Constraints.

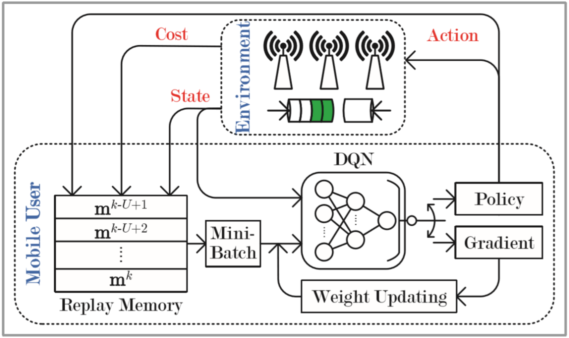
Deep Reinforcement Learning

Model:



Deep Reinforcement Learning

Method:



My work

$$\mathcal{P}: \Phi^{\star} = \operatorname*{argmax}_{\Phi} \mathbb{E}_{\Phi}\left[(1-\gamma)\sum_{t=1}^{\infty}\gamma^{t-1} r(\mathbf{x}^t, \Phi) | \mathbf{x}^1 = \mathbf{x}\right], \forall \mathbf{x} \in \mathcal{X},$$

where $r(\mathbf{x}^t, \Phi)$ is defined by

$$\begin{aligned} r(\mathbf{x}^t, \Phi) &= \sum_{i\in\mathcal{N}} \Delta_i^t - \varrho \cdot \Bigg(\phi \cdot \sum_{j\in\mathcal{M}} \big(\mathbb{1}\{I_s^t = j\} \\ &+ \mathbb{1}\{I_d^t = 1\}\big) + \zeta \cdot \sum_{i\in\mathcal{N}} \mathbb{1}\{Q_i^t \leq 0\} \\ &+ \xi \cdot \sum_{i\in\mathcal{N}} E_i^t(s) \cdot \big(\mathbb{1}\{s \notin \mathcal{S}_i^t \wedge E_i^t(s) \neq 0\}\big)\Bigg), \end{aligned}$$

where Δ_i^t is defined as

$$\Delta_i^t = \sum_{s\in\{s'\in\mathcal{S} | A_{i,s'}^t = 1\}} \Big(\mathbb{1}\{I_s^t = j^{\star}\} \cdot \big(b_{i,c}^t(s) - b_{i,e}^t(s, j^{\star})\big)\Big).$$

My Questions

How to Design *Novel* Models with Methods Embedded Naturally?

⊵ Fullfill understanding on *Convex Optimization*?

⊵ Stop Compromising of system model for *fancy* mathematical derivation?

⊵ Find *Stream* and *Tide*?

Outline

1. Application of Optimization Methods
   - ▷ Existing Methods and Their Applications
   - ▷ How to Design Novel Models with Methods Embedded Naturally?

2. Edge AI
   - ▷ Existing Paradigms
   - ▷ Preparation for Designing `Egent`

Existing Paradigms

▷ **Use Machine Learning Methods to solve traditional *resource management* or *latency minimization* problems**

▷ **Design new structure with heterogeous edge sites for AI-enabled apps**

1. Edge Federated Learning

Federated Learning

Federated Learning features ***distributed learning***† at edge devices and ***model-update aggregation***‡ at an edge server.

2. Learning-driven Communication

Learning-driven Communication

Conventional philosophy in traditional wireless communication

The traditional design objectives of wireless communications, i.e., ***coomunication reliablity*** and ***data-rate maximization***, do not directly match that of edge learning.

$\Downarrow$ A conceptual change

Learning-driven communication

The ***coupling*** between communication and learning in edge learning systems should be exploited.

A more detailed slide on *Learning-driven Communication* can be found at ▸ Link

Preparation for designing Egent

Network models compactable for Egent

Knowledge on Communication in this paper greatly enlightens me the desgin compactable network/communication models, which is a bottomed layer of Egent.

- Noise in training-data transmission maybe is not that important.
- Long-term observations and collected data on user profiles can be utilized for joint resource manegment and model-training.
- Collaboration between cloud and edge learning can be in-depth studied.
- Mobility management of users not only effects the offloading decisions, but also incurs frequent handovers among edge servers (not service migration).