Dynamic Content Updates in Heterogeneous Wireless Networks

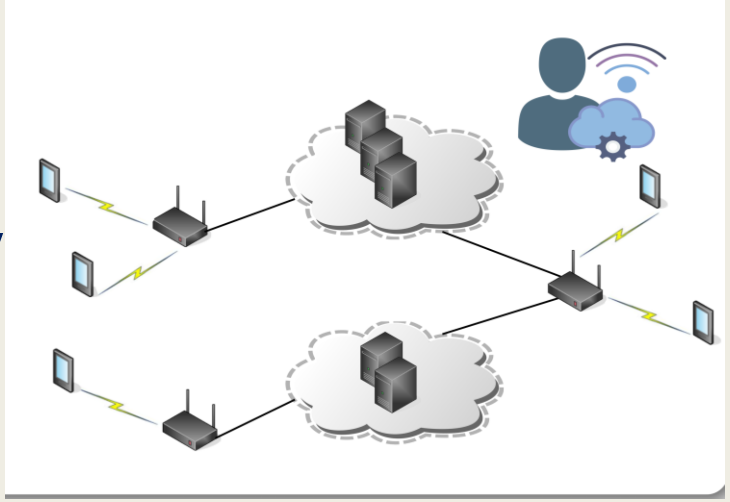
Mehdi S. H. Abad, Emre Ozfatura, Ozgur Ercetin, Deniz Gunduz

Sabancı University, Istanbul, Turkey

Imperial College London, United Kingdom

Funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement Tactilenet No 690893.

Edge Caching



- ❑ Caching at edge nodes help improve user quality
  - o Popular contents stored and served from small cell base stations (SBSs).
  - o Reduces backhaul cost and delay.

- ❑ Many solutions exists:
  - o Uncoded vs. coded caching
    - o K. Shanmugam et al., “FemtoCaching: Wireless content delivery through distributed caching helpers”, IEEE Transactions on Information Theory, 2013.
    - o K. Poularakis, “Approximation Algorithms for Mobile Data Caching in Small Cell Networks”, IEEE Transactions on Communications, 2014
  - o Known vs. unknown content popularity
    - o A. Sadeghi et al., “Optimal and scalable caching for 5g using reinforcement learning of space-time popularities”, IEEE Journal of Selected Topics in Signal Processing, 2018
    - o P. Yang et al., “Content popularity prediction towards location-aware mobile edge caching”, IEEE Transactions on Multimedia, 2018

Dynamic content edge caching

Most Internet content is dynamic.

- E.g., news, weather forecast, social networking videos, virtual reality games.
    - Providing personalized experience for users based on the time dependent events.
    - Event driven gaming experience in MMORPG games.
- Stale content reduces user satisfaction.
- Dynamic content in caches should be updated frequently.
- Updating edge caches frequently burdens backhaul!

Motivation

- ❑ Dynamic contents makes the ***age*** of contents important for the users:
  - *Refreshing the contents frequently will maximize the QoE of the users but would also increase the backhaul cost.*
  - *Refreshing the contents rarely, will reduce the backhaul cost but would degrade the user QoE.*
- ❑ A smart content refreshment strategy is required for striking a balance between the QoE of users and the backhaul cost.

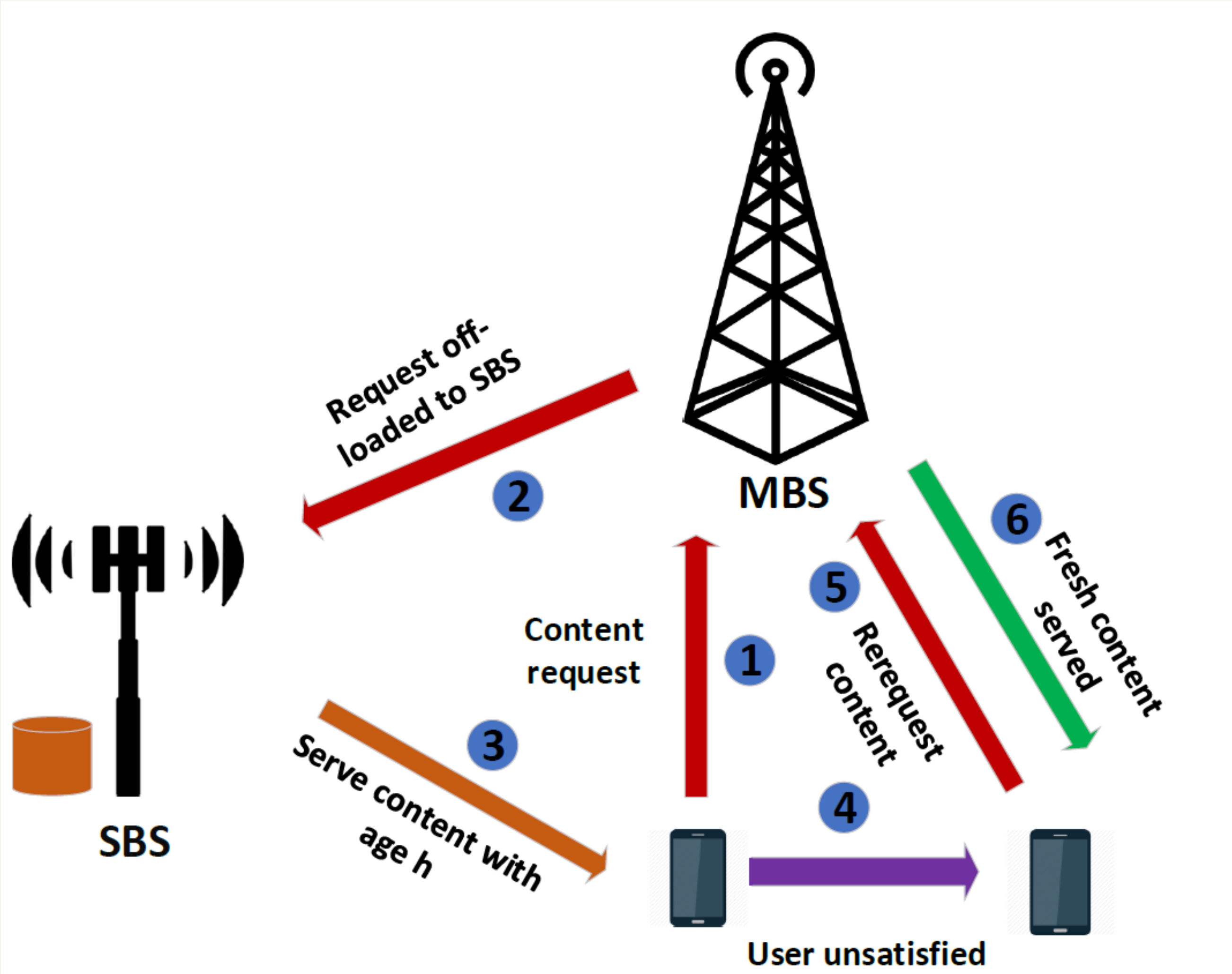
- ❑ User preference for varying age of content is different.
- ❑ Updating contents requires accessing costly backhaul link.
- ❑ User preference is UNKNOWN.

Approach

- We formulate the problem as an infinite-horizon Markov decision process (MDP)
- Show that MDP is separable to reduce the exponentially large state space.
- *Show that the optimal policy has threshold structure on the age of the contents.*
- We formulate and solve the problem of finding the optimal thresholds by a multi armed bandit problem.

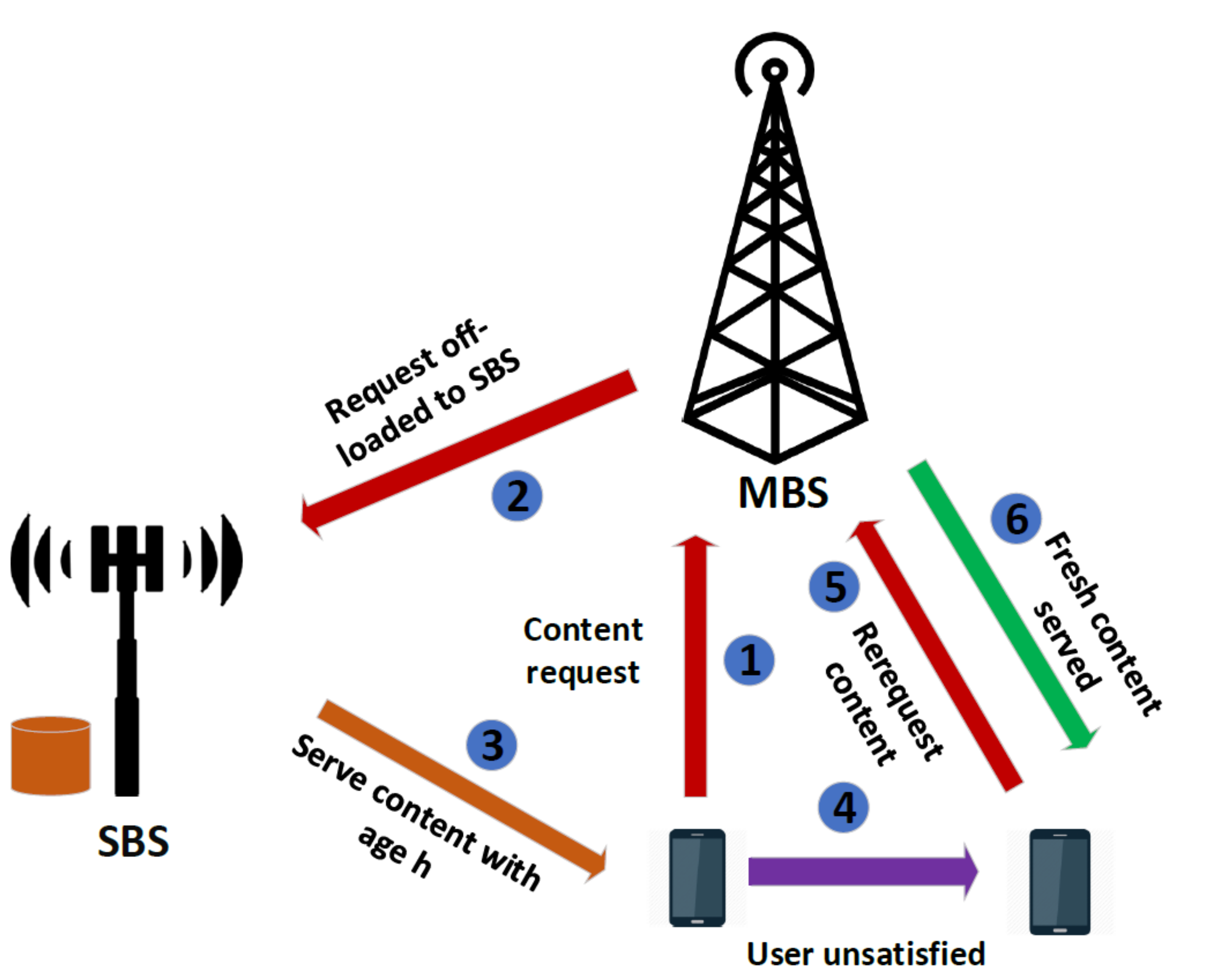
System model

- Consider N popular contents
- **Parameters:**
  - p_n: popularity of content n.
  - $P^n_{redirect}(h_n)$: probability of redirecting content n, when its age is h_n.
- **State:**
  - $h_n(t)$: age of content n at time t
- **Action:**
  - $\boldsymbol{d}(t) = \left(d_1(t), \ldots, d_N(t)\right)$: decision vector at time t
    - $d_n(t) = 1$: update content n
    - $d_n(t) = 0$: do not update content n



Users’ request model

1. User request a content from macro base station (MBS)
2. MBS offloads user’s request to SBS for service
3. SBS serves the requested content with age of h
4. The service process is finished *if* user is satisfied with the age of content; *else:*
5. *User makes another request from SBS*
6. *MBS serves the user with fresh content*



Problem Formulation

- $\lambda(t)$: Total number of users arriving to network at time t
- $\lambda_{rn}(t)$: number of requests for content n, redirected to MBS
  - $\lambda_{rn}(t)$ is governed by the age $h_n(t)$ as well as popularity
- $C\left(\lambda_{r1}(t), \dots, \lambda_{rN}(t), \boldsymbol{d}(t)\right)$: cost of serving redirected requests when action is $\boldsymbol{d}(t)$ and age is $\boldsymbol{h}(t)$
  - Linear backhaul cost:

$$C\left(\lambda_{r1}(t), \dots, \lambda_{rN}(t), \boldsymbol{d}(t)\right) = \sum_{n=1}^{N} C_{\mathrm{n}}(\lambda_{rn}(t), d_{\mathrm{n}}(t))$$

  - $C_n(\lambda_{rn}(t), d_n(t)) = C_n(\lambda_{rn}(t), 0) + d_n(t) C_{BH}$
    - $C_n(\lambda_{rn}(t), 0)$: cost of $\lambda_{rn}(t)$ number of users redirected to MBS.
    - C_{BH}: Backhaul cost associated with updating a content
- We aim at minimizing the expected cost over infinite horizon by optimizing the decision vector $\boldsymbol{d}(t)$:

$$\min_{\boldsymbol{d}(t)} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} C\left(\lambda_{r1}(t), \dots, \lambda_{rN}(t), \boldsymbol{d}(t)\right)$$

Optimal policy

- $\boldsymbol{h}(t) = \big(h_1(t), \cdots, h_N(t)\big)$: state of the system at time t

$$h_n(t+1) = \min\{\big(1 - d_n(t)\big)(h_n(t) + 1), h_{max}\}$$

- h_{max}: Corresponds to an age when the content becomes stale.
- $P(\boldsymbol{\acute{h}}|\boldsymbol{h}, \boldsymbol{d})$: transition probability, $\boldsymbol{\acute{h}} \to \boldsymbol{h}$ when action $\boldsymbol{d}$ is taken at state $\boldsymbol{h}$
- $V(\boldsymbol{h})$: value function at state $\boldsymbol{h}$

  $V_{\boldsymbol{d}}(\boldsymbol{h})$: action-value function at state $\boldsymbol{h}$ when action $\boldsymbol{d}$ is taken
- Bellman equations:
  - $V_{\boldsymbol{d}}(\boldsymbol{h}) = \bar{C}(\boldsymbol{h}, \boldsymbol{d}) + P(\boldsymbol{\acute{h}}|\boldsymbol{h}, \boldsymbol{d})\ V(\boldsymbol{\acute{h}})$
  - $\bar{C}(\boldsymbol{h}, \boldsymbol{d}) = \boldsymbol{E}\big(C(\lambda_{r1}, \ldots, \lambda_{rN}, \boldsymbol{d})\big)$
- Expectation is w.r.t PDF of λ_{rn}

Bellman optimality criteria dictates:

$$d^*(t) = \mathrm{arg}min_{\boldsymbol{d}} V_{\boldsymbol{d}}(\boldsymbol{h}(t))$$

Discussion

- State space of the MDP:
  - *Number of states increase exponentially with N*
  - *There are h_{max}^{N} number of states. h_{max} is the maximum age of a content.*
- Action space of the MDP:
  - *Number of actions is also exponential in N*
  - *There are 2^{N} number of actions*

MDP suffers from the curse of dimensionality!

Convergence becomes slow

Implementation requires a look-up table with a large size

Structure of the optimal policy

- Value function separable

$$V(\boldsymbol{h}) = V^{(1)}(h_1) + \cdots + V^{(N)}(h_N)$$

- $V^{(n)}(h_n)$: *value function for content* n
- *Follows from the linearity of cost function*
- Hence, each content can be considered separately
- *Bellman equations for content* n *becomes*
- $V_{d_n}^{(n)}(h_n) = \bar{C}_n(h_n, d_n) + d_n V^{(n)}(0) + (1 - d_n)V^{(n)}(h + 1)$

Theorem: There exists a threshold H_n on the age of each stored content at which it is optimal to update content n.

Separability of value function

$+$

optimality of threshold policy

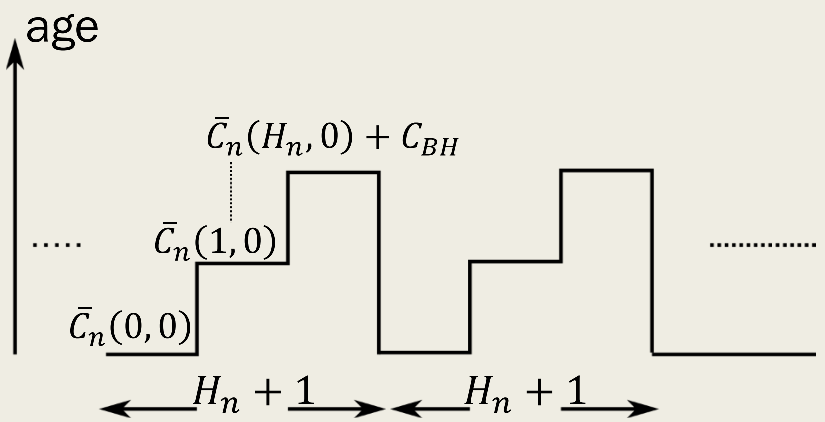
$\downarrow$

considerably simplify the procedure to find optimal decisions

Objective under threshold structure

- Objective function for each content n becomes

$$\min_{H_n} \frac{\sum_{h=0}^{H_n} \bar{C}_n(h,0)+C_{BH}}{H_n+1}$$



Need for a learning framework

Redirection probabilities are unknown

$\bar{C}_n(h,0)$ cannot be analytically computed.

Even if every other statistics is known

Even if cost function is calculated, the objective function is hard to solve

Learning optimal thresholds: Multi armed bandit (MAB) formulation

- MAB: For each content n, there is an agent:
  - *Agent chooses an arm (age update threshold, H_n)*
  - *Observes the random cost,* $\hat{C}(H_n) = \frac{\sum_{h_n=0}^{H_n} C_n(\lambda_{rn}(h_n),0)+C_{BH}}{H_n+1}$
  - Repeats the process until optimal arm, H_n^* is found

MAB agent

- $a_n \in \{0, \dots, h_{max}\}$: action of agent n
- $q^{(n)}(a_n)$: how favorable is to select threshold a_n for content n

ϵ-greedy algorithm

- Optimal action is chosen according to

$$a_n^* = \arg\min_{a_n} q^{(n)}(a_n)$$

  – *First, we need to determine the values of $q^{(n)}(a_n)$*

ϵ-greedy

With probability ϵ

chose a random arm

exploration improves the estimates of non-greedy actions' values

With probability $1-\epsilon$

Choose $\arg\min_{a_n} q^{(n)}(a_n)$

exploitation is favorable when we reach sufficiently reliable estimates of action values

take action a_n

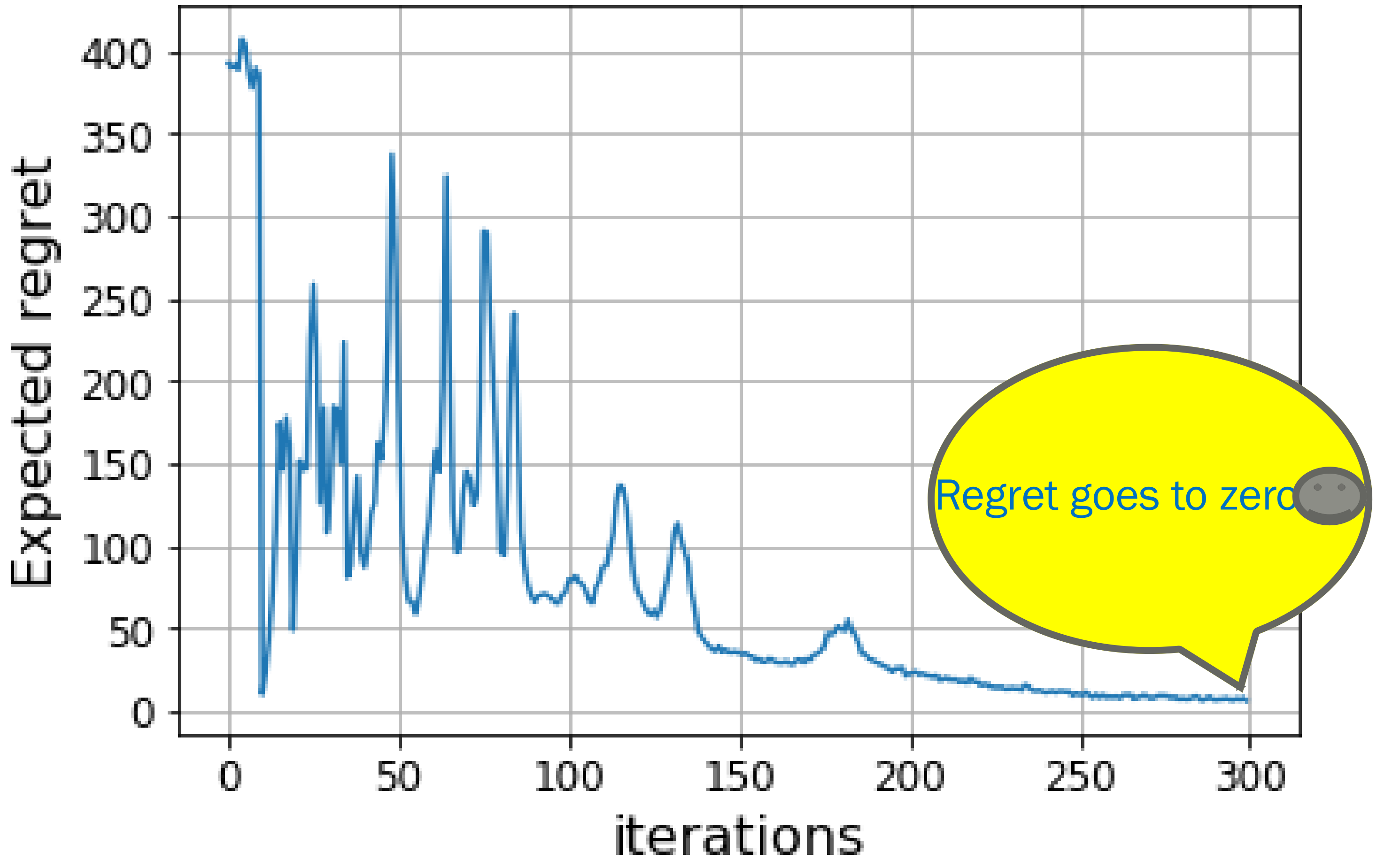
Observe sample mean $\hat{C}(a_n)$

$$q^{(n)}(a_n) \leftarrow q^{(n)}(a_n) + \alpha\left(\hat{C}(a_n) - q^{(n)}(a_n)\right)$$

α is the learning rate

Numerical Results

- $\lambda(t) \sim Poisson(100)$
- Popularity profile; $p \sim Zipf(2)$
- 5 applications
- $h_{max} = 9$
- $P_r(h) = 1 - e^{-0.2h}$,
- $C(x) = 10x$
- $C_{BH} = 500$



Expected regret: cost of the policy learned by MAB agent – optimal cost

Conclusions and future work

- Balancing user QoE and backhaul cost of updating dynamic content caches
  - Updating frequently results in higher backhaul cost
- Showed that the MDP is separable: reduced the state and action spaces
- Showed the optimal policy is of threshold type in age
  - Used multi-armed bandit framework to find efficient learning algorithms
- Future work
  - Non-linear cost functions
  - Random backhaul condition and constraint
  - Energy harvesting SBS