An Invitation to Research in Information Theory

Andrew Thangaraj

Department of Electrical Engineering
Indian Institute of Technology Madras
Email: andrew@ee.iitm.ac.in

Joint Work with Rahul Vaze (TIFR Mumbai),
Radhakrishna Ganti, Arijit Mondal (IIT Madras)
Thanks to Gerhard Kramer, Georg Böcherer (TU Munich)
Navin Kashyap (IISc Bangalore)

Outline of Talk

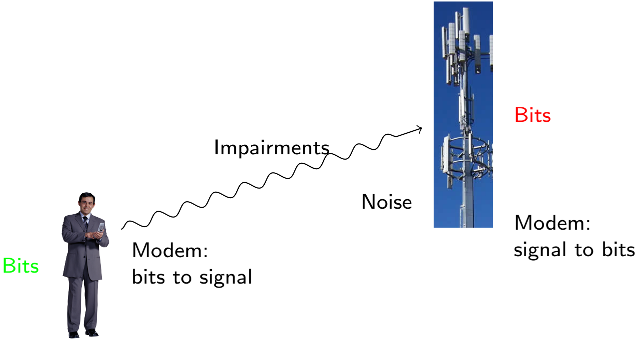
1. Information Theory for Communication Systems
   - Background, Current Status and Research
   - Noisy Runlength Constrained Channels
   - 1-bit Quantized ISI Channels
2. Online Algorithms for Basestation Allocation
   - Secretary problem
   - Basestation allocation and graph matching
   - Performance studies and comparison
3. Information Theory Research in India
   - Activities
   - IIT Madras

Communications, Information Theory and Coding Theory

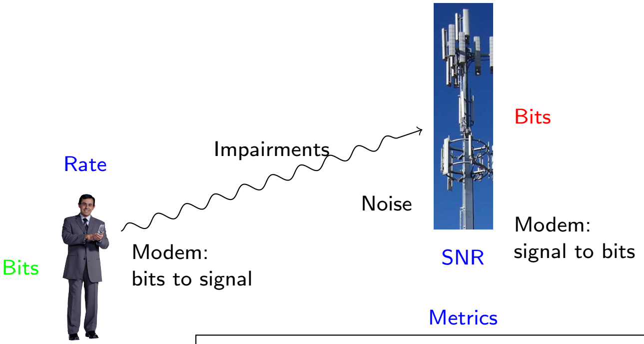
Bits

Modem:
bits to signal

Communications, Information Theory and Coding Theory

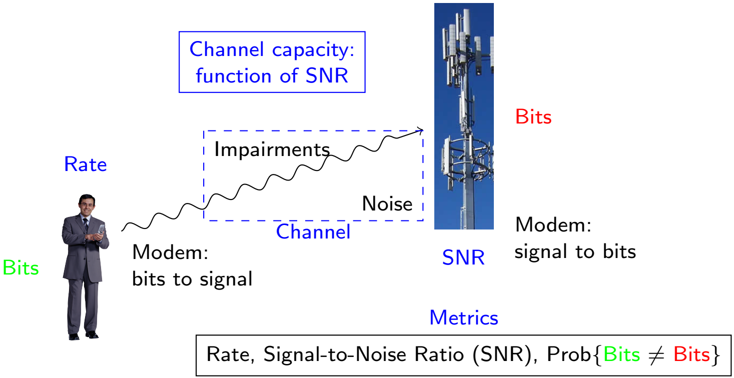


Communications, Information Theory and Coding Theory



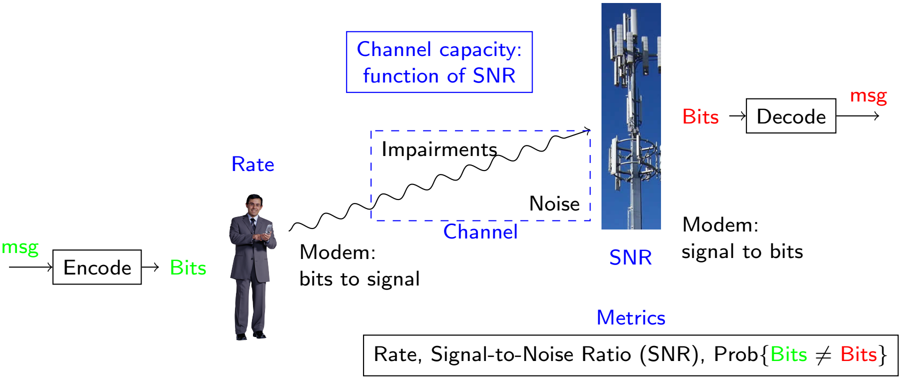
Rate, Signal-to-Noise Ratio (SNR), Prob{Bits $\neq$ Bits}

Communications, Information Theory and Coding Theory



- Information theory: model channel and compute capacity
  - If Rate $<$ Capacity, Prob{Bits $\neq$ Bits} can be as small as desired

Communications, Information Theory and Coding Theory



- Information theory: model channel and compute capacity
  - If Rate < Capacity, Prob{Bits ≠ Bits} can be as small as desired
- Coding theory: encoders and decoders to approach capacity

Research in Information Theory

Point-to-point communications

- Most capacity problems solved
- Performance close to capacity
- Channels with memory open

Multiterminal communications

- Most problems are open
- Practical systems far from limits

Emerging areas: beyond communications

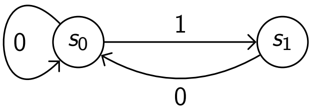
- Models and limits of learning
- Models and limits for storage

Runlength Constraint: No Consecutive 1s

Constrained binary sequence

Sequence of bits with no consecutive 1s

- Extension to (d, k)-constrained sequence used in recording channels (Marcus *et al* '98)
- Represented as a walk in a state diagram



How many such sequences?

$A_N =$ **Number of length-N binary vectors with no consecutive 1s**

- $N = 2$: 00, 10, 01 - $A_2 = 3$
- $N = 3$: 000, 001, 010, 100, 101 - $A_3 = 5$
- $N = 4$: 0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010 - $A_4 = 8$

$$\boxed{\text{For large } N\text{: } A_N \approx \left(\frac{\sqrt{5}+1}{2}\right)^N}$$

- $\dfrac{1}{N}\log_2 A_N \to \log_2 \dfrac{\sqrt{5}+1}{2}$

How many such sequences?

$A_N =$ **Number of length-N binary vectors with no consecutive 1s**

- $N = 2$: 00, 10, 01 - $A_2 = 3$
- $N = 3$: 000, 001, 010, 100, 101 - $A_3 = 5$
- $N = 4$: 0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010 - $A_4 = 8$

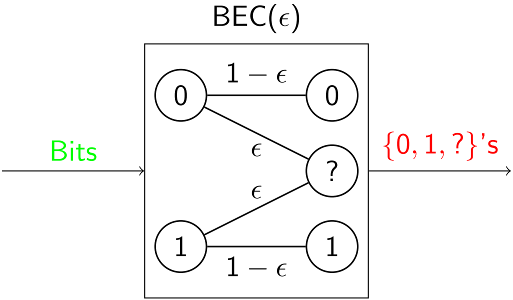
$$\boxed{\text{For large } N\text{: } A_N \approx \left(\frac{\sqrt{5}+1}{2}\right)^N}$$

- $\dfrac{1}{N}\log_2 A_N \to \log_2 \dfrac{\sqrt{5}+1}{2}$
- What happens when such a sequence is transmitted across a channel?

Some Prior Work

- Number of constrained sequences
  - C. E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., 1948.
- Lower bounds for constrained BSC
  - E. Zehavi and J. K. Wolf, "On runlength codes," IEEE Trans. Inf. Theory, 1988.
- Lower and upper bounds
  - D. M. Arnold, H. A. Loeliger, P. O. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," IEEE Trans. on Inf. Theory, 2006.
  - P. Vontobel and D. Arnold, "An upper bound on the capacity of channels with memory and constraint input," in IEEE ITW 2001.
  - Y. Li and G. Han, "Input-constrained erasure channels: Mutual information and capacity," in IEEE ISIT 2014.
- Feedback capacity for erasure channel (which serves as upper bound)
  - O. Sabag, H. H. Permuter, and N. Kashyap, "The feedback capacity of the binary erasure channel with a no-consecutive-ones input constraint," IEEE Trans. on Inf. Theory, Jan 2016.

Binary Erasure Channel (BEC) Model

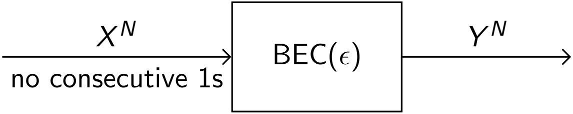


Elias '54

Capacity: $1-\epsilon$ bits per channel use

- Every bit gets erased independently with probability ϵ
- Simple model that captures the essence of the problem

Noisy Constrained Channels

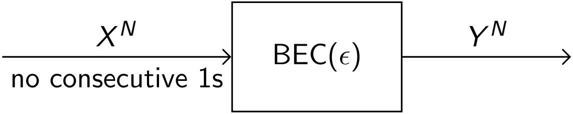


$C(\epsilon)$: **Capacity of constrained channel**

$$\boxed{C(\epsilon) = \lim_{N\to\infty} \frac{1}{N} \max_{p_{X^N}:\, X^N \in \mathcal{X}_{d,k}^N} I(X^N; Y^N)}$$

- Capacity of constrained noisy channels: has proven to be difficult to characterize or bound even for simple channels

Noisy Constrained Channels



$C(\epsilon)$: **Capacity of constrained channel**

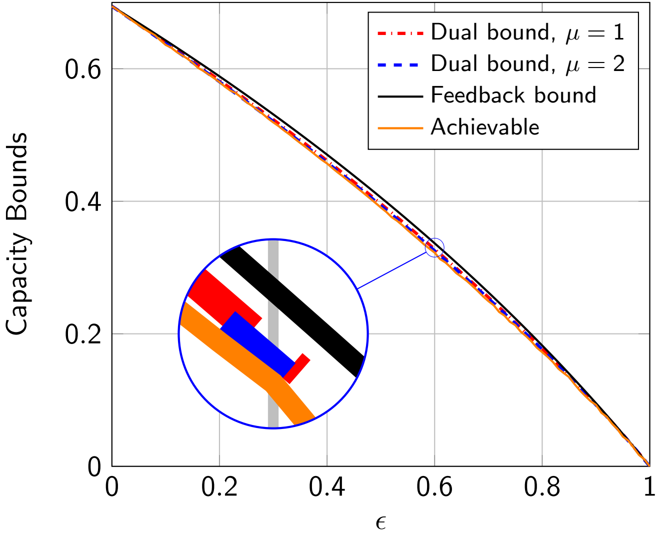
$$\boxed{C(\epsilon) = \lim_{N\to\infty} \frac{1}{N} \max_{p_{X^N}:\, X^N \in \mathcal{X}_{d,k}^N} I(X^N; Y^N)}$$

- Capacity of constrained noisy channels: has proven to be difficult to characterize or bound even for simple channels

[A. Thangaraj, "Dual Capacity Upper Bounds for Noisy Runlength Constrained Channels," presented at IEEE Information Theory Workshop 2016]

Dual bound:
$$\boxed{\begin{gathered} C(\epsilon) \leq (1-\epsilon)^2 \log_2(1/\beta) + \epsilon(1-\epsilon)\log_2(2-\beta), \\ \beta \in (0,1] \text{ solves } \beta^{2(1-\epsilon)} = 1-\beta \end{gathered}}$$

Plot: Binary Erasure Channel



Dual bound: This work, Feedback bound: Sabag *et al* '16, Achievable: Arnold *et al* '06

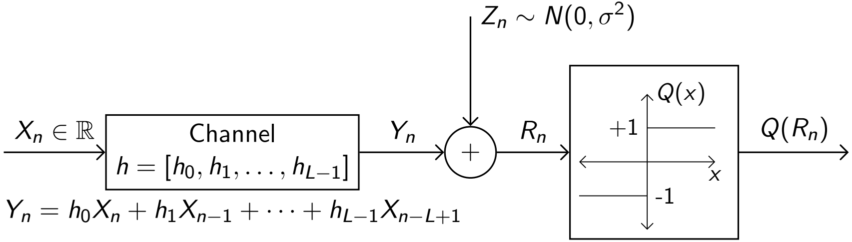
Remarks

Main technique: Dual Capacity Upper Bound

- Dual capacity bound is useful in scenarios where characterizing the exact capacity is difficult.
  - Related work: A. Thangaraj, G. Kramer, and G. Böcherer, "Capacity upper bounds for discrete-time amplitude-constrained AWGN channels," in IEEE ISIT 2015, https://arxiv.org/abs/1511.08742.
- In my work, dual capacity bound was used for noisy constrained channels
  - (d, k)-constrained binary symmetric channel and binary-input Gaussian channels
- Future work
  - Dual bounds for other channels with memory, such as the Inter Symbol Interference (ISI) channel

Additional details: https://arxiv.org/abs/1609.00189

ISI Channel with 1-bit Quantized Output



- Continuous input, 1-bit quantized output (why this model?)
- Average power constraint: $E[\|X\|^2] \leq NP$

Problem: How to signal?

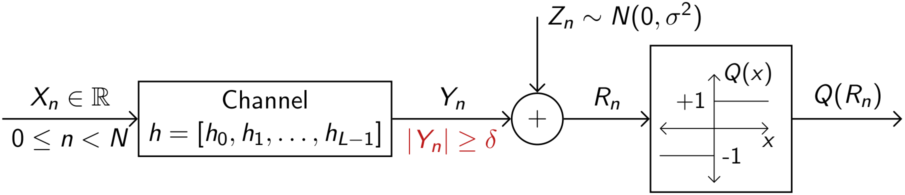
- ISI channel with continuous output: Inner equalizer + Outer code
- ISI channel with 1-bit output: How to equalize?

Prior Work vs This Work

- Why ISI with continuous input, quantized output?
  - Millimeter-wave [Sun *et al* '14], SERDES [Harwood *et al* '07]
- Prior work
  - ISI with continuous input/output: [Hirt-Massey '88], [Shamai-Ozarow-Wyner '91], [Shamai-Laroia '96]
  - Non-ISI quantized output: [Singh-Dabeer-Madhow '09]
  - ISI, optimize quantizer at receiver: [Zeitler-Singer-Kramer '12]
  - Millimeter wave, 1-bit quantized output: [Mo-Heath '14]
- This work: Equalizers (transmitter block precoding) for ISI channels with continuous input, 1-bit quantized output

Threshold Precoder and Noisefree Approximation

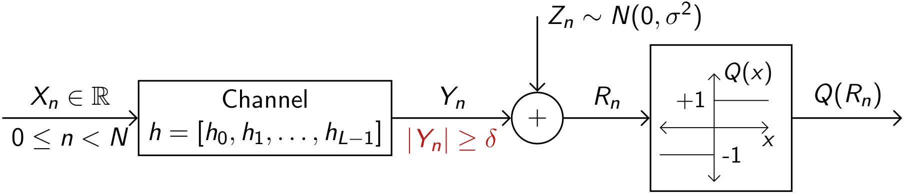
δ-threshold precoding for 1-bit quantized ISI channel



- Choose $\{X_n\}$ with threshold constraint: $|Y_n| = |\sum_{i=0}^{L-1} h_i X_{n-i}| \geq \delta$

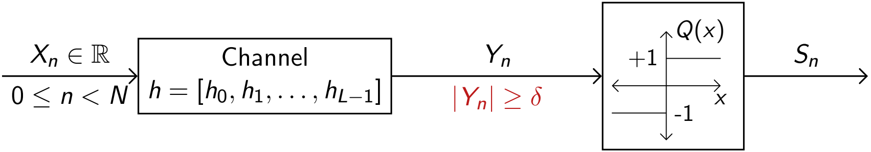
Threshold Precoder and Noisefree Approximation

δ-threshold precoding for 1-bit quantized ISI channel



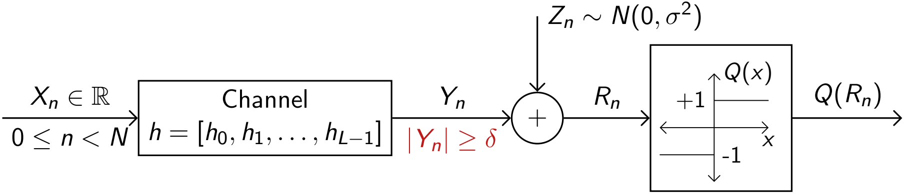
- Choose $\{X_n\}$ with threshold constraint: $|Y_n| = |\sum_{i=0}^{L-1} h_i X_{n-i}| \ge \delta$

Noisefree approximation of threshold-precoded channel



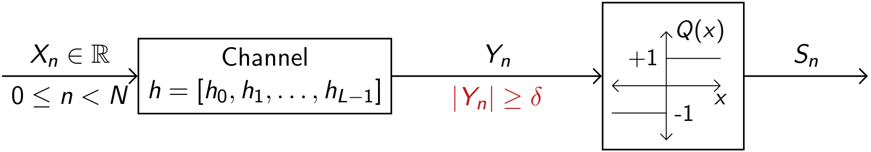
Threshold Precoder and Noisefree Approximation

δ-threshold precoding for 1-bit quantized ISI channel



- Choose $\{X_n\}$ with threshold constraint: $|Y_n| = |\sum_{i=0}^{L-1} h_i X_{n-i}| \geq \delta$

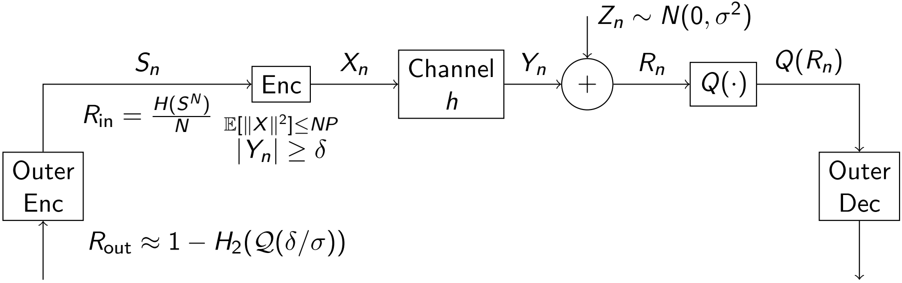
Noisefree approximation of threshold-precoded channel



$$\boxed{\text{Approximation error: } \mathbb{P}(S_n \neq Q(R_n)) \leq \mathcal{Q}(\delta/\sigma)}$$

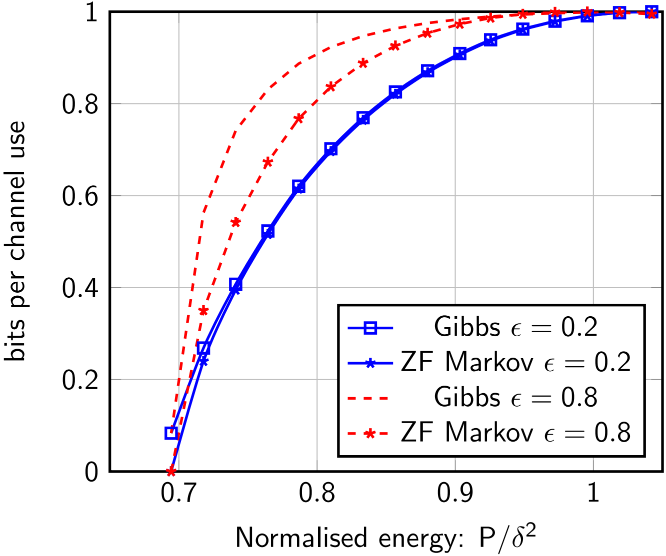
Signaling Method: Precoder + Outer Code

Precoder + Outer code for the noisy channel

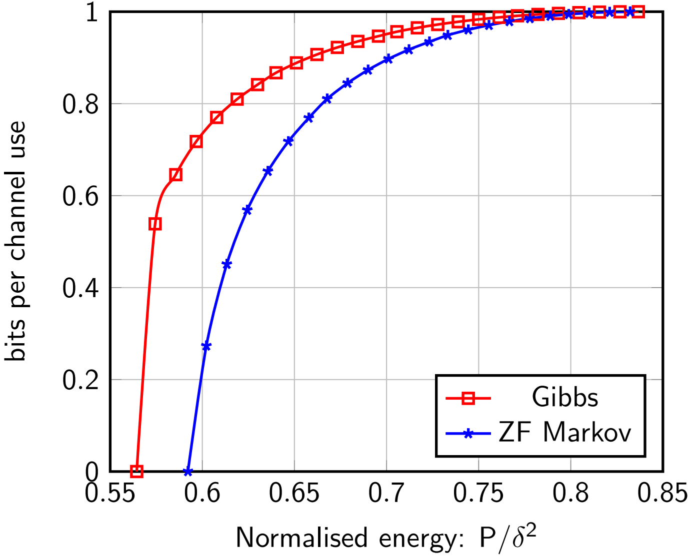


[R. Ganti, A. Thangaraj and A. Mondal, "Approximation of Capacity for ISI Channels with One-bit Output Quantization," IEEE ISIT 2015.]

Numerical Example I: Channel [1 ϵ]



Numerical Example II: Channel [-0.3, 1, 0.6]



Remarks

- Signaling method for ISI channels with 1-bit ouput
  - Precoder + Outer error control code
- Design of precoder
  - Threshold precoding
  - Use of noisefree approximation, Gibbs sampling: optimal
- Achievable schemes
  - Zero-forcing Markov: good practical method
- Future
  - Several applications in 5G and mmwave
  - Extensions to MIMO

1. Information Theory for Communication Systems
   - Background, Current Status and Research
   - Noisy Runlength Constrained Channels
   - 1-bit Quantized ISI Channels

2. Online Algorithms for Basestation Allocation
   - Secretary problem
   - Basestation allocation and graph matching
   - Performance studies and comparison

3. Information Theory Research in India
   - Activities
   - IIT Madras

The Secretary Problem: "Online" Interview

Goal: Select candidate with best score or weight

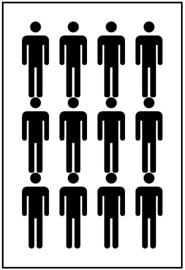
(C1) Interview one at a time

(C2) Decide online: select or reject

(C3) Rejected cannot be called back

(C4) No more interviews after selection

n candidates

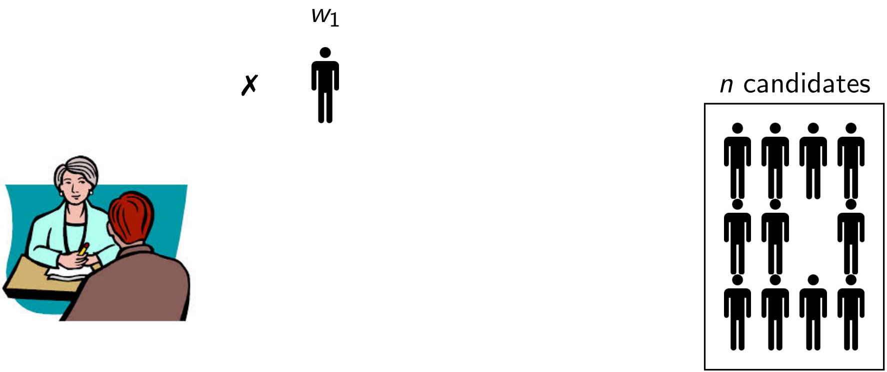


The Secretary Problem: "Online" Interview

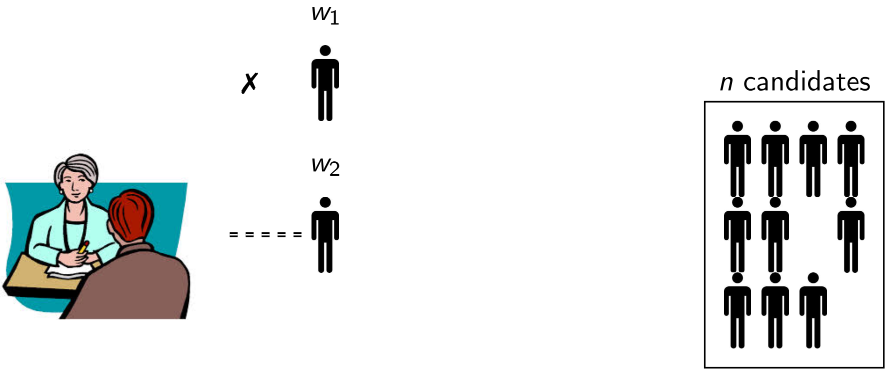
n candidates

w_1

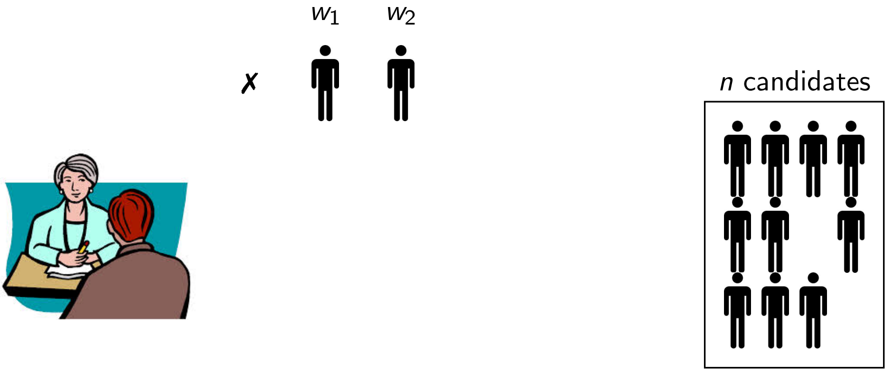
The Secretary Problem: “Online” Interview



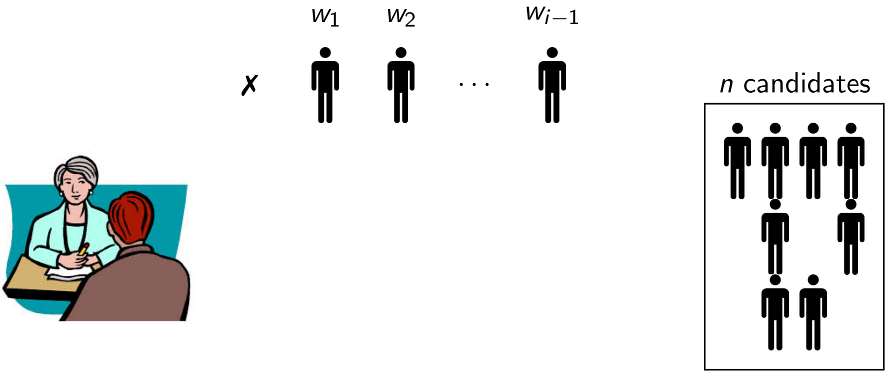
The Secretary Problem: "Online" Interview



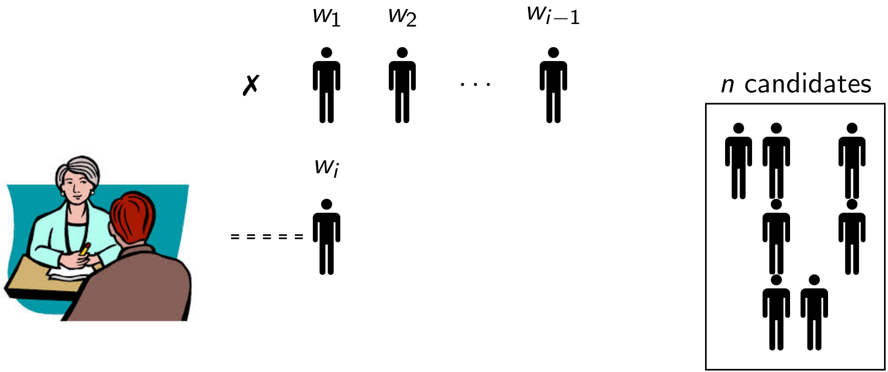
The Secretary Problem: “Online” Interview



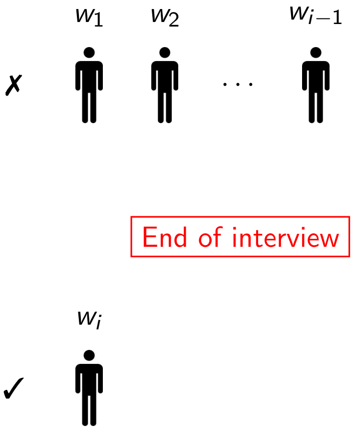
The Secretary Problem: “Online” Interview



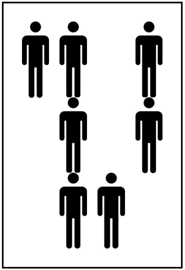
The Secretary Problem: “Online” Interview



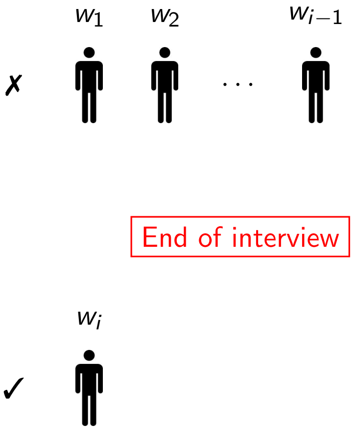
The Secretary Problem: "Online" Interview



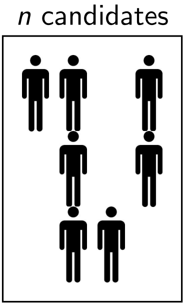
n candidates



The Secretary Problem: "Online" Interview



End of interview

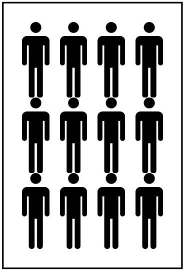


Central question: How good is online compared to offline?

Offline interview always hires applicant with maximum score. Can an online interview do the same?

Model for Candidates and Order of Interview

n candidates

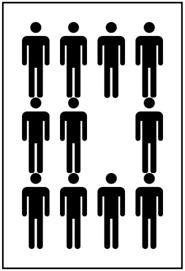


- Fixed n and weights: $w_1, w_2, \ldots, w_n$

Model for Candidates and Order of Interview

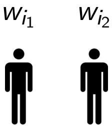
w_{i_1}

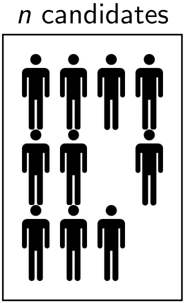
n candidates



- Fixed n and weights: $w_1, w_2, \ldots, w_n$
- Order of interview is uniformly random

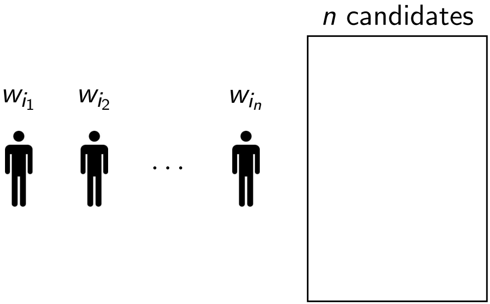
Model for Candidates and Order of Interview





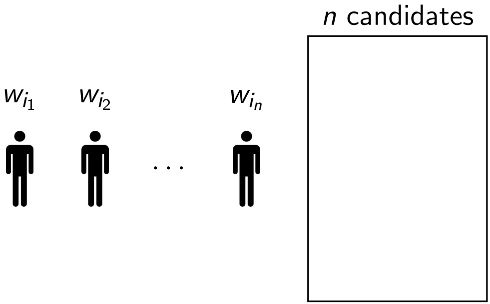
- Fixed n and weights: $w_1, w_2, \ldots, w_n$
- Order of interview is uniformly random

Model for Candidates and Order of Interview



- Fixed n and weights: w_1, w_2, ..., w_n
- Order of interview is uniformly random
- $\{i_1, i_2, \ldots, i_n\}$: permutation of $\{1, 2, \ldots, n\}$, uniformly random

Model for Candidates and Order of Interview



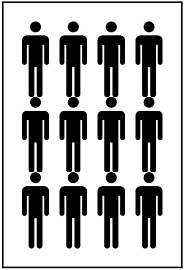
- Fixed n and weights: w_1, w_2, ..., w_n
- Order of interview is uniformly random
- $\{i_1, i_2, \ldots, i_n\}$: permutation of $\{1, 2, \ldots, n\}$, uniformly random

Offline (knows permutation) vs Online

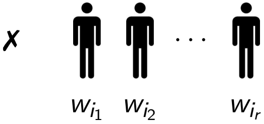
In how many of the $n!$ permutations, can an online interview select the candidate with maximum weight?

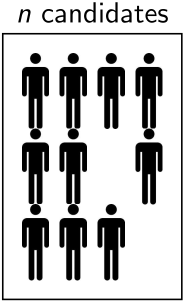
Online Algorithm $\mathcal{A}(r)$ for the Secretary Problem

n candidates



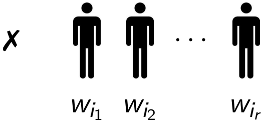
Online Algorithm $\mathcal{A}(r)$ for the Secretary Problem

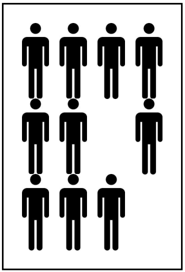




- Reject first r candidates

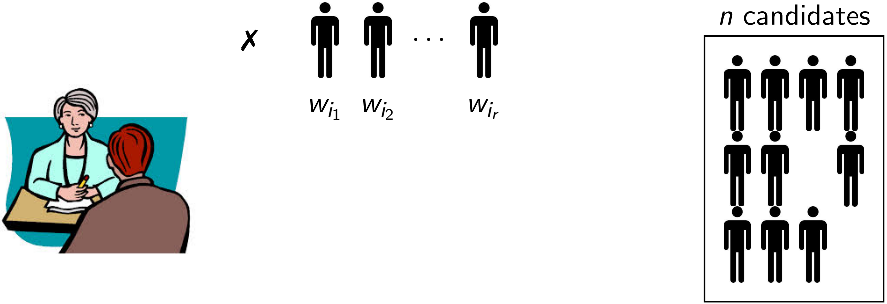
Online Algorithm $\mathcal{A}(r)$ for the Secretary Problem





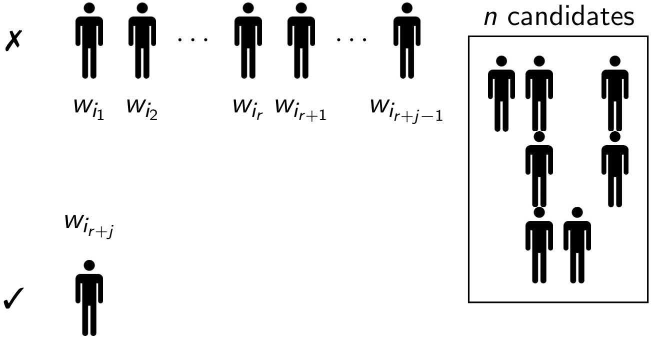
- Reject first r candidates
- Find best weight, so far: $T = \max(w_{i_1}, w_{i_2}, \ldots, w_{i_r})$

Online Algorithm $\mathcal{A}(r)$ for the Secretary Problem



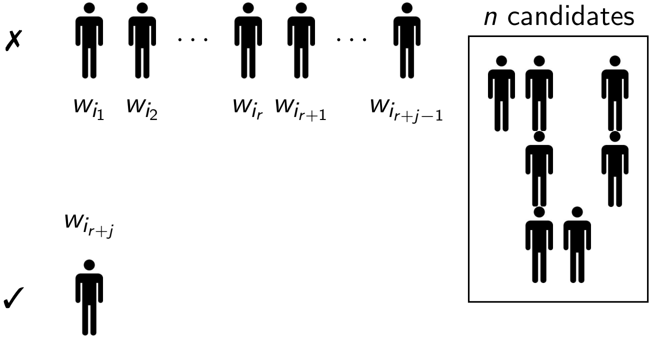
- Reject first r candidates
- Find best weight, so far: $T = \max(w_{i_1}, w_{i_2}, \ldots, w_{i_r})$
- From subsequent candidates, select first one with weight $\geq T$

Online Algorithm $\mathcal{A}(r)$ for the Secretary Problem



- Reject first r candidates
- Find best weight, so far: $T = \max(w_{i_1}, w_{i_2}, \ldots, w_{i_r})$
- From subsequent candidates, select first one with weight $\geq T$

Online Algorithm $\mathcal{A}(r)$ for the Secretary Problem



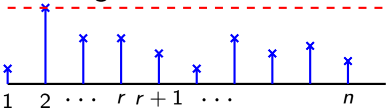
- Reject first r candidates
- Find best weight, so far: $T = \max(w_{i_1}, w_{i_2}, \ldots, w_{i_r})$
- From subsequent candidates, select first one with weight $\geq T$
- If there is none, select last candidate

Offline versus Online $\mathcal{A}(r)$

- Offline algorithm
  - Knows entire sequence of weights ahead of time: w_{i_1}, w_{i_2}, ..., w_{i_n}
  - Suppose m-th candidate has maximum weight w_{i_m}
  - Reject candidates 1 to $m-1$
  - Select candidate m
- Online algorithm $\mathcal{A}(r)$
  - Surprise! In nearly 36% of the cases, $\mathcal{A}(r)$ will select candidate m
  - Best choice for r: $r = n/e$
  - Fraction of successes is independent of n

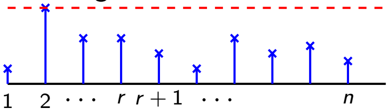
Analysis of Online Algorithm $\mathcal{A}(r)$

- Max weight candidate comes before r: not selected

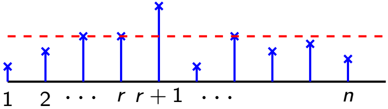


Analysis of Online Algorithm $\mathcal{A}(r)$

- Max weight candidate comes before r: not selected

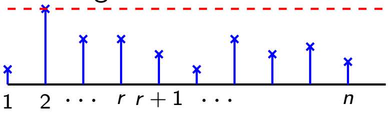


- Max weight candidate at $r+1$: selected always

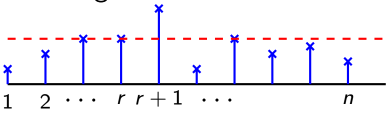


Analysis of Online Algorithm $\mathcal{A}(r)$

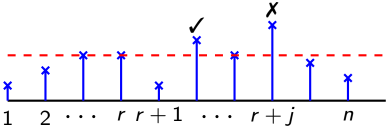
- Max weight candidate comes before r: not selected

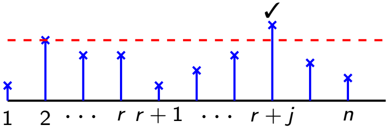


- Max weight candidate at $r+1$: selected always



- Max weight candidate at $r+j$: selected with some probability



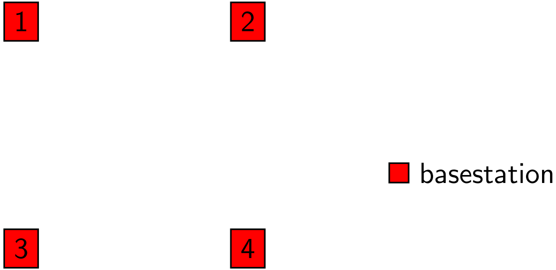


Prob: $r/(r+j-1)$

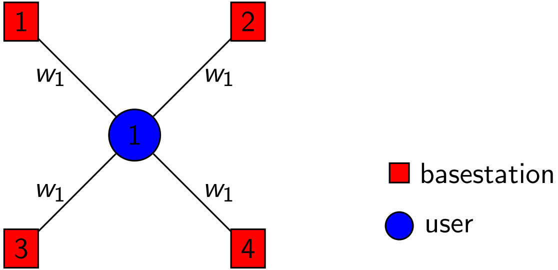
Probability Calculation

$$
\begin{aligned}
&\Pr\{\text{Selecting Top Applicant}\} \\
&= \sum_{j=1}^{n-r} \Pr\{\text{Top applicant in } r+j\} \Pr\{\text{Best in first } r+j-1 \text{ within } r\} \\
&= \sum_{j=1}^{n-r} \frac{1}{n} \;\; \frac{r}{r+j-1} = \frac{r}{n} \sum_{i=r}^{n-1} \frac{1}{i} \\
&\approx \frac{r}{n} \log \frac{n}{r}, \text{ maximised by choosing } r = n/e \\
&= \frac{1}{e} \approx 0.36.
\end{aligned}
$$

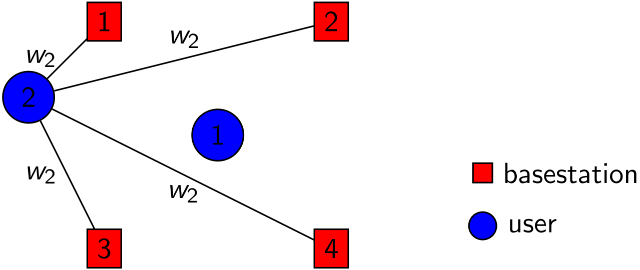
Basestation Allocation Problem



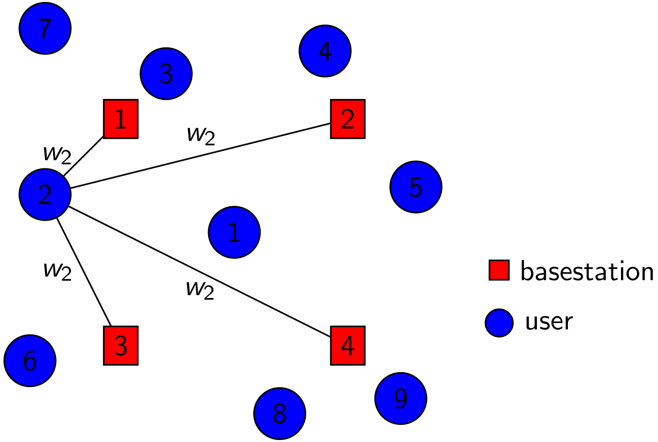
Basestation Allocation Problem



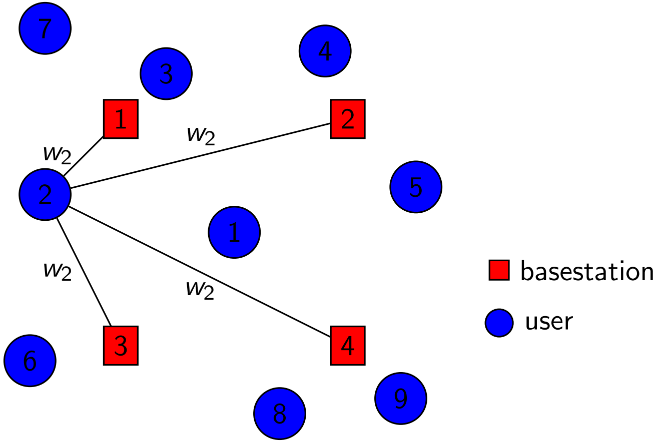
Basestation Allocation Problem



Basestation Allocation Problem



Basestation Allocation Problem

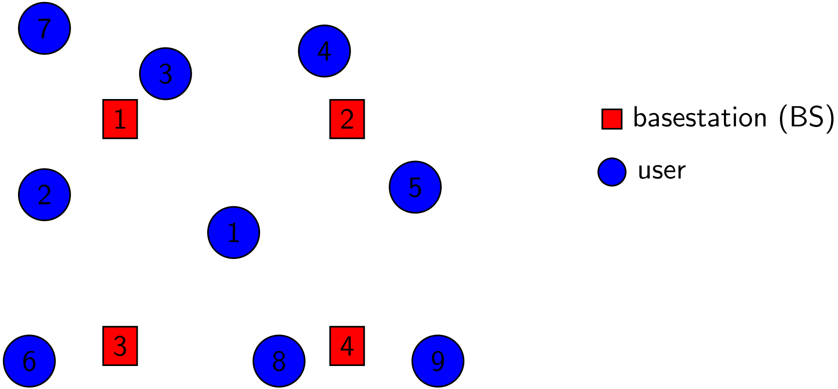


Basestation allocation

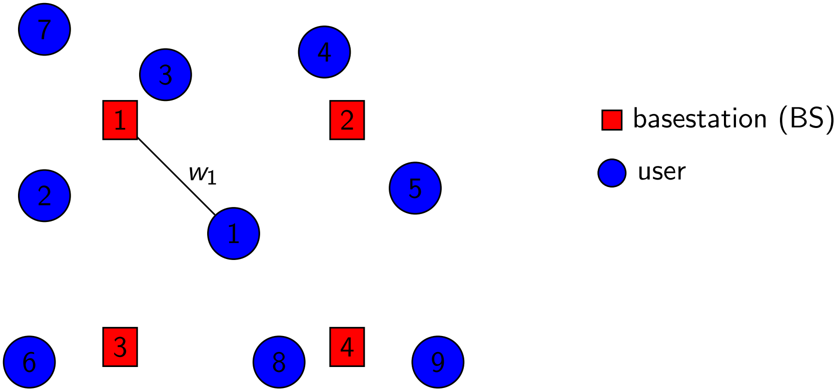
Allot n users to one out of m basestations

w_i: weight of user i to all basestations

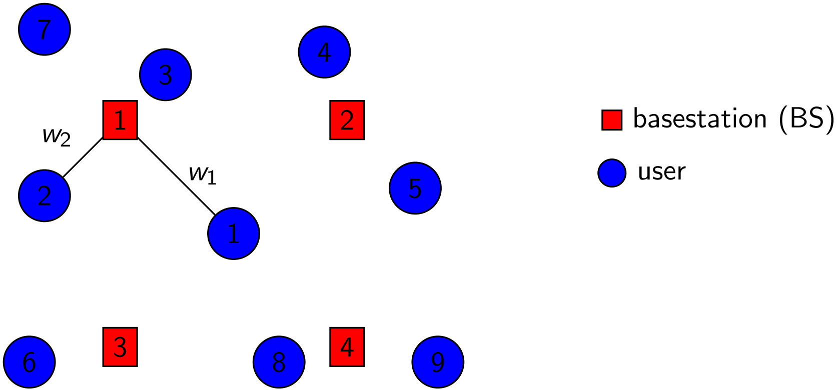
Time-sharing Utility at the Basestation



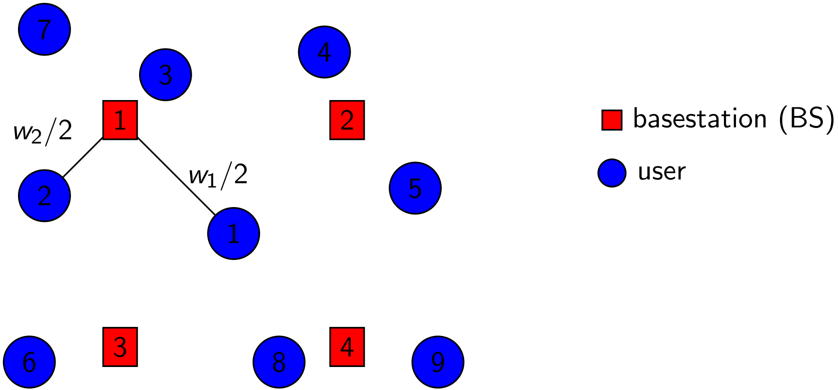
Time-sharing Utility at the Basestation



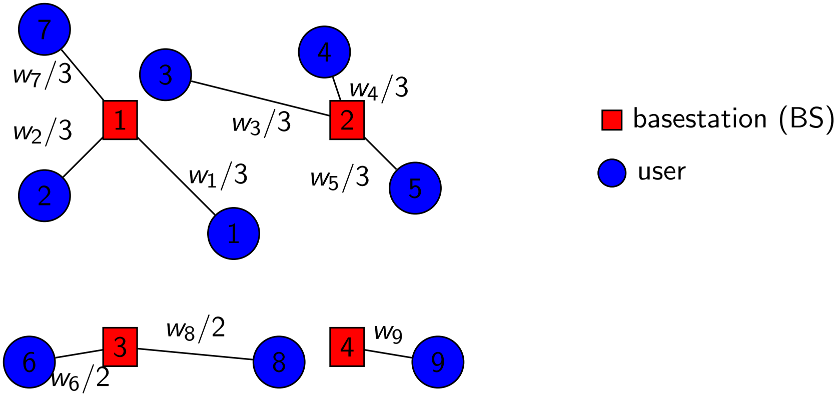
Time-sharing Utility at the Basestation



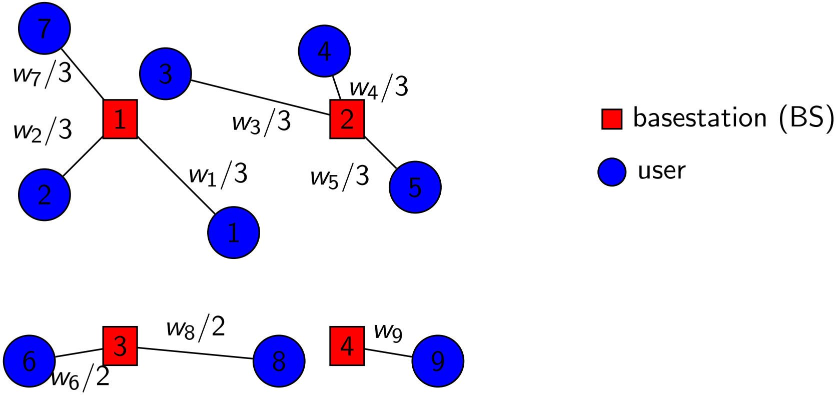
Time-sharing Utility at the Basestation



Time-sharing Utility at the Basestation



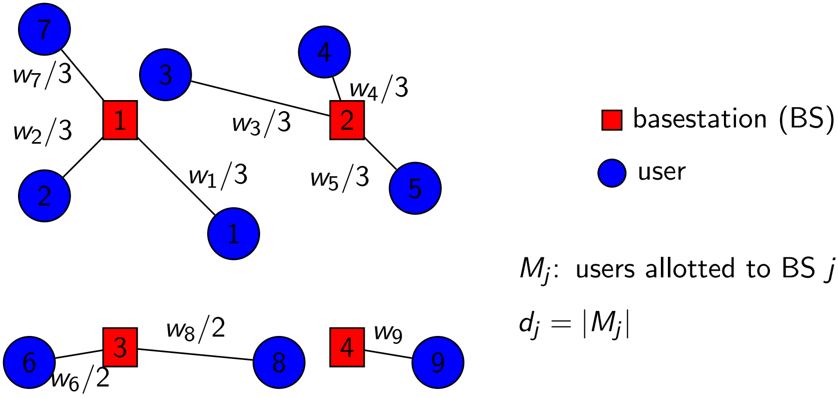
Time-sharing Utility at the Basestation



Time-sharing utility: Total rate from all basestations

$$\frac{w_1 + w_2 + w_7}{3} + \frac{w_3 + w_4 + w_5}{3} + \frac{w_6 + w_8}{2} + w_9$$

Time-sharing Utility at the Basestation



M_j: users allotted to BS j

$d_j = |M_j|$

Time-sharing utility: Total rate from all basestations

$$\left(\frac{\sum_{i\in M_1} w_i}{d_1}\right) + \left(\frac{\sum_{i\in M_2} w_i}{d_2}\right) + \cdots$$

Basestation Allocation: Offline vs Online

Goal of allocation: maximize time-sharing utility

Find M_1, M_2, ... such that $\left(\frac{\sum_{i\in M_1} w_i}{d_1}\right) + \left(\frac{\sum_{i\in M_2} w_i}{d_2}\right) + \cdots$ is maximized.

- Offline
  - sequence of user weights $\{w_i\}$ is known before arrival
  - what is the best offline algorithm?
- Online
  - weight of i-th user w_i is revealed at time i
  - user i needs to be allotted a basestation at time i
  - allotment of user i cannot be changed later

Main question: Fix n and weights $w_1, w_2, \ldots, w_n$

Users arrive in uniformly random order. How good are online algorithms compared to offline?

Optimal Offline Algorithm

How to maximize time-sharing utility?

$$\left(\frac{\sum_{i\in M_1} w_i}{d_1}\right) + \left(\frac{\sum_{i\in M_2} w_i}{d_2}\right) + \cdots$$

Intuition: Allot high weight users to basestations with low d

Optimal Offline Algorithm

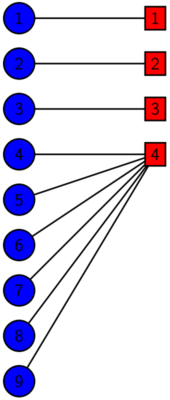
Optimal offline allocation: n users to m basestations

- Find top $m-1$ users
- Allot them individually to $m-1$ basestations
- Assign other $n-m+1$ users to basestation m
- More details and proof: http://arxiv.org/abs/1308.1212

Optimal Offline Algorithm

Optimal offline allocation: n users to m basestations

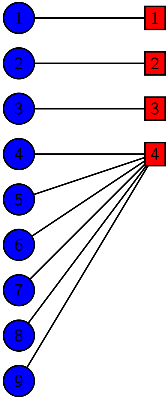
- Find top $m-1$ users
- Allot them individually to $m-1$ basestations
- Assign other $n-m+1$ users to basestation m
- More details and proof: http://arxiv.org/abs/1308.1212



Optimal Offline Algorithm

Optimal offline allocation: n users to m basestations

- Find top $m-1$ users
- Allot them individually to $m-1$ basestations
- Assign other $n-m+1$ users to basestation m
- More details and proof: http://arxiv.org/abs/1308.1212



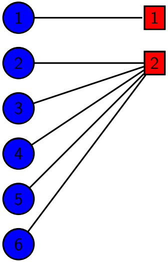
$$\text{Best utility} = w_1 + w_2 + w_3 + \frac{1}{n-3}(w_4 + \cdots + w_n)$$

- Assume: $w_1 > w_2 > \cdots > w_n$, $m = 4$

From Secretary Problem to Basestation Allocation

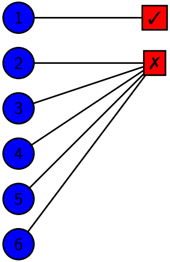
Optimal offline allocation: $m = 2$ basestations

- Find max weight user
- Assign max weight user to basestation 1
- Assign other $n - 1$ users to basestation 2



- Secretary problem: select candidate with max weight online
- Basestation allocation: identify max-weight user online

From Secretary Problem to Basestation Allocation



Optimal offline allocation: $m = 2$ basestations

- Find max weight user
- Assign max weight user to basestation 1
- Assign other $n-1$ users to basestation 2

- Secretary problem: select candidate with max weight online
- Basestation allocation: identify max-weight user online

Online algorithm for $m = 2$ basestations

Run the algorithm $\mathcal{A}(r)$ for the secretary problem to select max weight user for basestation 1 (change: do not stop after selection)
Allot all rejected users to basestation 2

Online Algorithm $\mathcal{A}_m(r)$ for m Basestations

Algorithm $\mathcal{A}_m(r)$: Extension of $\mathcal{A}(r)$ to find top $m-1$ users

1. Allocate the first r test users to basestation m, and compute the $(m-1)$-th best weight, denoted T, among the first r users.
2. For $i > r$, if weight $\leq T$,
   1. Allocate user i to basestation m.
3. For $i > r$, if weight $> T$,
   1. Allocate user i to basestations 1 though $m-1$ in a round robin fashion.
   2. Update T to be the $(m-1)$-th best weight seen so far.

How good is $\mathcal{A}_m(r)$?

For $2 \leq m \leq 20$, Optimized r: $\approx 0.22n$, and

$$\text{Expected } \frac{\text{Online utility}}{\text{Offline utility}} \geq 0.46$$

Sketch of Computations

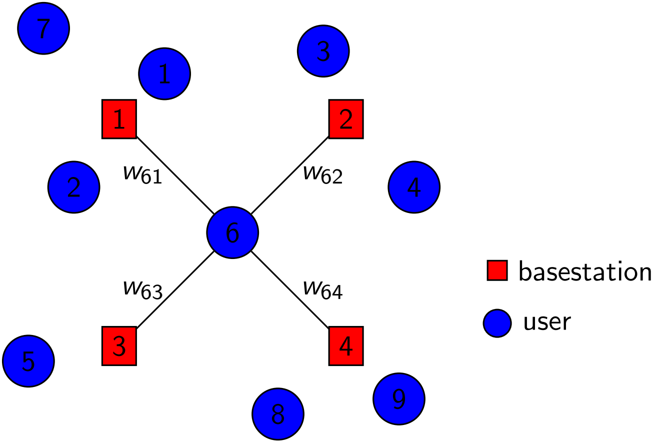
- Basestation m: used for dumping "bad" users
- Basestation 1 through $m-1$: "good" users
- Main computation: $\Pr\{S=d\}$, where S: number of good users

$$\mathbb{P}(S=d) = \sum_{\substack{r+1 \le i_1 < i_2 < \\ \cdots < i_d \le n}} \left(\prod_{i \in \{i_1, i_2, \ldots, i_d\}} \frac{m-1}{i} \right) \left(\prod_{\substack{r+1 \le i \le n \\ i \notin \{i_1, i_2, \ldots, i_d\}}} 1 - \frac{m-1}{i} \right)$$

$$\to \left(\frac{r}{n}\right)^{m-1} \frac{1}{d!} \left((m-1) \log_e \frac{n}{r} \right)^d$$

- Fact about permutations that is useful
  - π: uniformly random permutation, $b_i = |\{j : 1 \le j \le i, \pi(j) \ge \pi(i)\}|$
  - $b_i \sim \mathrm{unif}\{1, 2, \ldots, i\}$, independent of $b_{i'}$ for $i' \neq i$
  - More details in http://arxiv.org/abs/1308.1212

More Practical Scenario: Arbitrary Weights

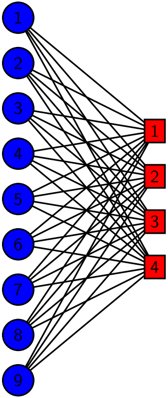


Arbitrary Weights

Each user has possibly different weights to each basestation: w_{ij}, $1 \leq j \leq m$

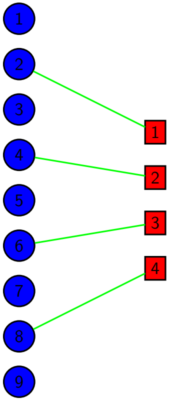
Max Weight Matching Problem

- Bipartite graph: G
  - Edge $e = (i, j)$ has weight $w_e = w_{ij}$



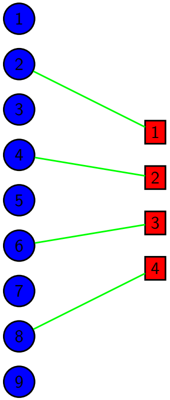
Max Weight Matching Problem

- Bipartite graph: G
  - Edge $e = (i, j)$ has weight $w_e = w_{ij}$
- Matching M in G:
  Set of non-intersecting edges
  - Weight of matching = sum of weights of edges in matching



Max Weight Matching Problem

- Bipartite graph: G
  - Edge $e = (i, j)$ has weight $w_e = w_{ij}$
- Matching M in G:
  Set of non-intersecting edges
  - Weight of matching = sum of weights of edges in matching



Max Weight Matching Problem

Find matching with maximum weight
Like "Secretary problem" for equal weights case, online Max Weight Matching is used for basestation allocation in the arbitrary weights case.

Arbitrary Weights: Online Algorithm

- Idea: Randomized online allocation algorithm using an online bipartite graph matching algorithm $\mathcal{A}_M$

HideandSeek Algorithm

- Let $j_0 \in \{1, 2, \ldots, m\}$ be chosen uniformly at random.
  G_{-j_0}: G with basestation j_0 deleted.
- Use online algorithm $\mathcal{A}_M$ to find the max-weight matching in the graph G_{-j_0}. Denote this matching as $\mathcal{A}_M(G_{-j_0})$.
- Allocate the users in the matching $\mathcal{A}_M(G_{-j_0})$ to the $m-1$ basestations other than j_0, while all other users are allocated to basestation j_0 itself.

Arbitrary Weights: Online versus Offline

Theorem

With randomized input the HideAndSeek algorithm has

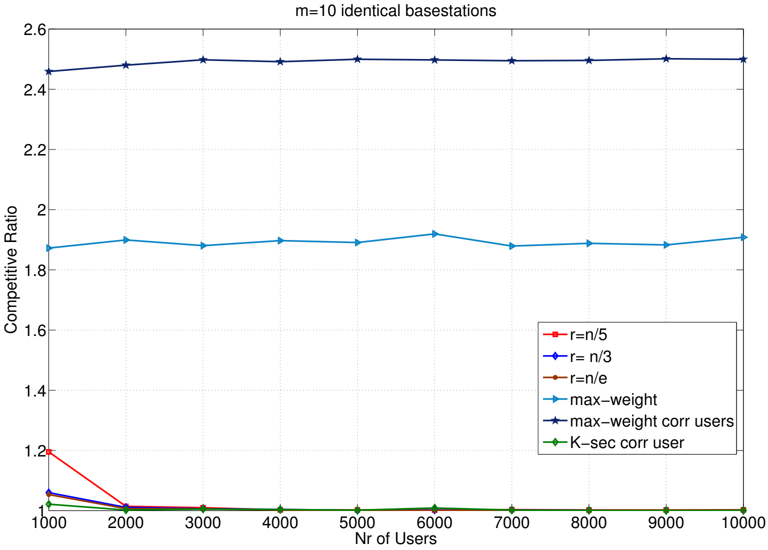
$$\textit{Competitive ratio} \triangleq \textit{Expected}\ \frac{\textit{Offline utility}}{\textit{Online utility}} \leq \frac{8m}{m-1}.$$

Proof.

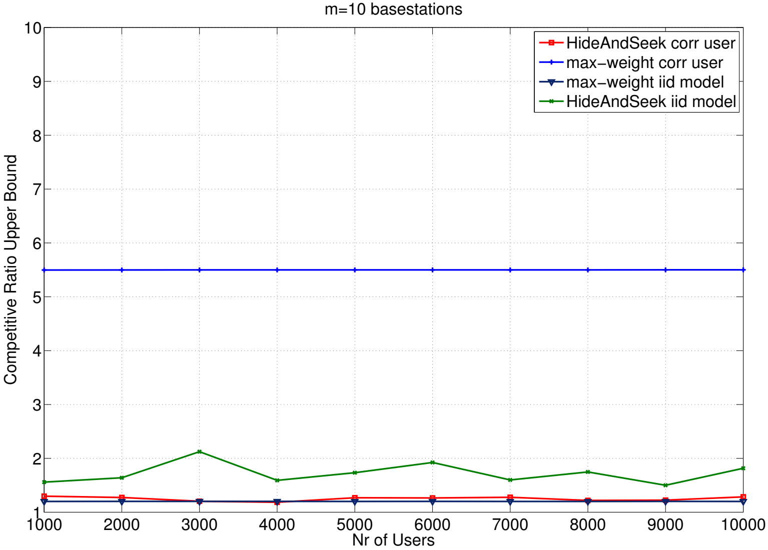
$$E[\text{weight}(\mathcal{A}_M(G_{-j_0}))] \geq \frac{m-1}{m} E[\text{weight}(\mathcal{A}_M(G))],$$

since j_0 is chosen uniformly at random.
Since $\mathcal{A}_M$ is within $1/8$ of the offline algorithm, the result follows. □

Simulation Results: Secretary-problem-based Algorithm



Simulation Results: HideAndSeek versus Max-weight



Summary Remarks

- Online algorithms
  - Good candidates for practical use.
  - Constant competitive ratio under random user arrivals.
  - Robust to statistical misspecification.
- Max-weight algorithms versus HideandSeek
  - Max-weight association has average-case competitive ratio n (can be shown).
  - The competitive ratio of HideAndSeek algorithm is $8m/(m-1)$.
  - In practice, under correlated user weights, HideandSeek outperforms max-weight.
- More details: http://arxiv.org/abs/1308.1212
- Extension: K. Thekumparampil, A. Thangaraj and R. Vaze, "Combinatorial Resource Allocation Using Sub-modularity of Waterfilling," IEEE Transactions on Wireless Communications, vol. 15, no. 1, pp. 206-216, January 2016.

1. Information Theory for Communication Systems
   - Background, Current Status and Research
   - Noisy Runlength Constrained Channels
   - 1-bit Quantized ISI Channels

2. Online Algorithms for Basestation Allocation
   - Secretary problem
   - Basestation allocation and graph matching
   - Performance studies and comparison

3. Information Theory Research in India
   - Activities
   - IIT Madras

Conferences and Schools

- National Conference in Communications (NCC)
  - Premier annual communications conference in the country
  - NCC 2017 is at IIT Madras
    - http://www.ncc2017.org/
    - Submission deadline: Oct 31, 2016
- Joint Telematics Group/IEEE Information Theory Society Summer School
  - Held annually at IIT Madras, IISc Bangalore or IIT Bombay
  - 8-hour courses on two or three topics
  - 2016L http://www.ece.iisc.ernet.in/ jtg/2016/
- Signal Processing and Communications Conference (SPCOM)
  - Held once in two years at IISc Bangalore
  - This year's conference website: http://ece.iisc.ernet.in/ spcom/2016/
- International Conference on Communication Systems and Networks (COMSNETS): http://www.comsnets.org/
- Premier international conferences in Information Theory
  - IEEE International Symposium on Information theory
  - IEEE Information Theory Workshop

Opportunities at IIT Madras

- MS/PhD programmes
  - https://research.iitm.ac.in/
  - Deadline for Jan-May 2017: October 24
  - Final year UG, PG students can apply
- Summer fellowships
  - https://sfp.iitm.ac.in/
- MoU between NITT and IITM
  - Facilitates student exchange and joint research
  - Top 10% of 3rd year undergraduate students
    - can apply for direct PhD programme at the end of 3rd year
    - if admitted, can spend final year at IIT Madras and then continue on to PhD