Repeated Games, Optimal Channel Capture, and Open Problems for Slotted Multiple Access

Michael J. Neely
University of Southern California
https://viterbi-web.usc.edu/~mjneely/

58th Allerton Conference on Communication, Control, and Computing, Sep. 28, 2022

Outline

1. MAC Game Competition (7 semesters at USC)
   - Winning algorithm
   - To be (Greedy) or Not To Be (Greedy)?
2. Minimizing expected time to capture a channel:
   - Exponentially growing **decision space**
   - Novel optimality proof for 2, 3, 4, 6 users

Part 1: EE 550 MAC Game Competition

- Two users compete for a channel
- Packet transmission = 1 slot
- Compete over 100 slots
- Binary decision on each slot: Transmit (1) or not (0)?
- Idle/Success/Collision
- Students submit algorithms in Matlab:
  (Base decision at time t on *history of prior decisions* of yourself and your opponent. Can use randomness)

Example 100 slot Game

	ALG 1	ALG 2	RESULT
1	?	?	
2			
3			
⋮	⋮	⋮	⋮
100			

Example 100 slot Game


	ALG 1	ALG 2	RESULT
1	1	1	
2			
3			
⋮	⋮	⋮	⋮
100			

Example 100 slot Game

	ALG 1	ALG 2	RESULT
1	1	1	***Collision***
2			
3			
⋮	⋮	⋮	⋮
100			

Example 100 slot Game


	ALG 1	ALG 2	RESULT
1	1	1	***Collision***
2	?	?	
3			
⋮	⋮	⋮	⋮
100			

Example 100 slot Game


	ALG 1	ALG 2	RESULT
1	1	1	***Collision***
2	0	0	
3			
⋮	⋮	⋮	⋮
100			

Example 100 slot Game


	ALG 1	ALG 2	RESULT
1	1	1	***Collision***
2	0	0	***Idle***
3			
⋮	⋮	⋮	⋮
100			

Example 100 slot Game

	ALG 1	ALG 2	RESULT
1	1	1	***Collision***
2	0	0	***Idle***
3	?	?	
⋮	⋮	⋮	⋮
100			

Example 100 slot Game

	ALG 1	ALG 2	RESULT
1	1	1	***Collision***
2	0	0	***Idle***
3	1	0	
⋮	⋮	⋮	⋮
100			

Example 100 slot Game

	ALG 1	ALG 2	RESULT
1	1	1	***Collision***
2	0	0	***Idle***
3	1	0	***ALG 1 gets 1 point***
⋮	⋮	⋮	⋮
100			

Competition rules

1. All *n* algorithm pairs compete over 100 slot games
2. Goal: Get highest **sum score** over all games you play
3. The algorithms you compete against include:
   - All student-designed algs (**including yourself**)
   - NeverTransmit
   - AlwaysTransmit
   - 4-state

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Totals
A1	0	0	0	0	0	0	0	0	0	0	0
A2	100	0	1	0	48.89	0	19.99	10.72	0	50.03	230.6444
A3	98.03	0	49.49	0	49.41	49.33	21.61	24.9	0	33.27	326.0706
A4	39.94	0	1	0	20.46	0	20.06	10.69	0	36.27	128.4562
A5	49.95	0	0.49	0	25.04	0	9.98	5.33	0	24.88	115.6895
A6	1	0	49.67	0	0.5	0	19.05	31.93	0	25.26	127.4265
A7	100	0	19.12	0.54	49.98	18.81	16.45	19.21	0.56	34.46	259.1658
A8	50.52	0	23.85	0.02	25.34	31.57	18.65	24.74	0.19	27.37	202.2952
A9	100	0	1	0	49.97	0	19.88	10.76	0	38.18	219.8058
A10	50.01	0	16.68	13.75	25.15	24.76	15.61	22.73	11.81	24.98	205.5207

Figures of merit for an algorithm

- SelfCompetition score α: What is your expected score when playing an independent version of yourself?
- NoCompetition score β: What is your expected score when playing NeverTransmit?
- HumanCompetition score γ: Simulated over 135 algs

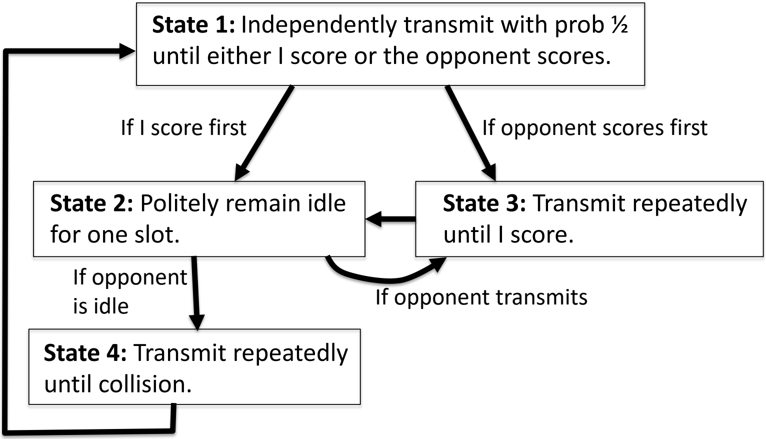
Def: A **deterministic** algorithm uses no rand() calls.

Lemma: Every deterministic algorithm has $\alpha = 0$.

Some baseline algs

- **AlwaysTransmit**
- **Tit-for-tat-1:**
  1. Slot 1: $X[1] = 1$
  2. Slot $t \in \{2, ..., 100\}$: $X[t] = X_{opponent}[t-1]$
- **Tit-for-tat-0:**
  Same as Tit-for-tat-1 except $X[1] = 0$.
- **4-state**
- **4-state with greedy ending**

4-state Alg



State 1: Independently transmit with prob ½ until either I score or the opponent scores.
If I score first
If opponent scores first
State 2: Politely remain idle for one slot.
State 3: Transmit repeatedly until I score.
If opponent is idle
If opponent transmits
State 4: Transmit repeatedly until collision.

Results of competition

	4-State	Second Place	AlwaysTransmit	AvgAlg
Fall 2021 (10 algs)	32.46	26.02	22.90	18.14
Fall 2020 (25 algs)	23.92	22.82	12.36	12.10
Fall 2019 (19 algs)	30.55	30.07	18.32	16.25
Spring 2018 (35 algs)	56.31	53.62	25.55	33.71
Fall 2018 (27 algs)	32.44	29.63	15.42	17.11
Spring 2017 (21 algs)	20.44	17.68	8.00	10.88
Fall 2016 (14 algs)	20.22	17.53	11.22	10.22

	SelfComp α	NoComp β	Tournament γ
4-state	49.500	98.000	24.613
Tit-for-tat-0	0	0	20.410
Tit-for-tat-1	0	1	15.326
AlwaysTransmit	0	100	10.714

- Scores are presented as average score per (100-slot) game.
- 4-state came in 1st place every semester
- AvgAlg is the average score over all algs that semester.

Theorem: 4-state gives optimal SelfCompetition score

Theorem:

a) The SelfCompetition score for 4-state is

$$\alpha = \frac{T-1}{2} + \left(\frac{1}{2}\right)^{T+1}$$

$T = 100 \implies \alpha \approx 49.500000000000000000000000000000394$

b) **(Converse):** No algorithm that competes against an independent copy of itself can do better.

Part 2: Expected time to capture channel

1. n users; slotted time
2. Everyone knows there are n
3. Users are indistinguishable (labels $\{1, 2, ..., n\}$ unknown)
4. Design an alg that is independently used by each user to minimize the expected time until the first success

Related work

- Distributed control
  1. Witsenhausen 1973, 1987
     Proof for $n = 3$ agents; $n > 3$ open
  2. Nayyar and Teneketzis 2019
     Common Information

- Regret-based and online convex opt
  1. Bubeck and Budzinski 2020
  2. Bubeck, Li, Peres, Sellke 2020
  3. Kalathil, Nayyar, Jain 2014

- Distributed MAC, Poisson arrivals, Splitting and Tree Algs
  1. Bertsekas and Gallager 1992
  2. Mosely and Humblet 1985
  3. Tsybakov and Mikhailov 1978, 1980, 1981
  4. Hayes 1978
  5. Capetanakis 1979

Collision feedback $F[t]$

At end of each slot t, all users receive feedback:

$$F[t] = \text{Number of users who transmitted}$$

- $F[t] = 0$ (Idle)
- $F[t] = 1$ (Success and done)
- $F[t] = 2$ (Collision of 2 users)
- $F[t] = 3$ (Collision of 3 users)

  ...
- $F[t] = n$ (Collision of n users)

We can know $F[t]$ by, for example,

1. Measuring energy in collision
2. Using bit signature and counting spikes in matched filter [Gollakota and Katabi ZigZag 2008, SigSag]

Proposed Alg for $n = 2$

Both users **independently transmit with prob** $1/2$ every slot until first success.

- Z = random time to first success.
- $z_2 = \mathbb{E}[Z]$
- $\boxed{z_2 = 2}$

Proposed Alg for $n = 3$

Transmit with prob p and observe $F[t]$:

- $F[t] = 0$:
- $F[t] = 1$:
- $F[t] = 2$:
- $F[t] = 3$:

Proposed Alg for $n = 3$

Transmit with prob p and observe $F[t]$:

- $F[t] = 0$: Repeat
- $F[t] = 1$:
- $F[t] = 2$:
- $F[t] = 3$: Repeat

Proposed Alg for $n = 3$

Transmit with prob p and observe $F[t]$:

- $F[t] = 0$: Repeat
- $F[t] = 1$: Success! (Done)
- $F[t] = 2$:
- $F[t] = 3$: Repeat

Proposed Alg for $n = 3$

Transmit with prob p and observe $F[t]$:

- $F[t] = 0$: Repeat
- $F[t] = 1$: Success! (Done)
- $F[t] = 2$: groups $\{\tilde{a}, \tilde{b}\}, \{\tilde{c}\} \implies$ Done in 1
- $F[t] = 3$: Repeat

Result for $n = 3$

- Get:

$$\mathbb{E}[Z] = \frac{1 + 3p^2(1-p)}{1 - p^3 - (1-p)^3}$$

- Now optimize p:

$$z_3 = \inf_{p \in (0,1)} \left\{ \frac{1 + 3p^2(1-p)}{1 - p^3 - (1-p)^3} \right\}$$

$$\implies p^* = 0.411972$$

$$\boxed{z_3 = 1.78795}$$

Proposed Alg for general n

Transmit with prob p and observe $F[t]$:

- $F[t] = 0$: Repeat
- $F[t] = 1$: Done in 1
- $F[t] = k \in \{2, ..., n-2\}$:

  Choose better of groups: $\{k$ users$\}, \{n-k$ users$\}$

- $F[t] = n-1$: Done in 2
- $F[t] = n$: Repeat

$$z_n = \inf_{p \in (0,1)} \left\{ \frac{1 + \sum_{i=2}^{n-1} \min\{z_i, z_{n-i}\} \binom{n}{i} p^i (1-p)^{n-i}}{1 - p^n - (1-p)^n} \right\}$$

Conjecture: This algorithm is optimal for all $n \in \{1, 2, 3, ...\}$
Have proof for special cases $n \in \{1, 2, 3, 4, 6\}$

Proof of converse for $n = 4$

- Consider **any algorithm** independently used by 4 users
- Let Z be random time to first success of this algorithm
- Want to show $\mathbb{E}[Z] \geq z_4$
- Idea: Consider **new system with 2 virtual users with enhanced capabilities!**
  (Can each send any number of packets per slot)
- Show virtual system has $\mathbb{E}[Z_{virtual}] \geq z_4$
- Show virtual system can emulate actual system
  (so $\mathbb{E}[Z] \geq \mathbb{E}[Z_{virtual}]$)

Conclusions

1. MAC Game
   - Sharing is good. Greedy is bad.
   - Randomness is required
   - 4-state consistently wins competitions
     (and maximizes self-score α)

2. Time to first capture
   - Complexity explosion in information state (and group state)
   - Interesting heuristic for all n
   - Optimality for $n \in \{1, 2, 3, 4, 6\}$
     (Novel method of virtual users with enhanced capabilities)

3. Open problems
   - $n = 5$; $n \geq 7$
   - Limited forms of feedback
   - Multiple channels

Related NSF projects

1. NSF SpecEES 1824418
   - M. J. Neely, "Repeated Games, Optimal Channel Capture, and Open Problems for Slotted Multiple Access," arXiv technical report, arXiv:2110.09638v1.
   - X. Zhou, I. Koprulu, A. Eryilmaz, M. J. Neely, "Low-Overhead Distributed MAC for Serving Dynamic Users over Multiple Channels," Proc. WiOpt 2021.
   - M. J. Neely, "Reversible Models for Wireless Multi-Channel Multiple Access," Proc. IEEE INFOCOM, 2021.
2. NSF CCF-1718477
   - M. J. Neely, "Fast Learning for Renewal Optimization in Online Task Scheduling," Journal of Machine Learning Research (JMLR) Sept. 2021.
   - M. J. Neely, "A Converse Result on Convergence Time for Opportunistic Wireless Scheduling," Proc. IEEE INFOCOM 2020.
   - K. Asgari and M. J. Neely, "Bregman-style Online Convex Optimization with Energy Harvesting Constraints," Proc. ACM Meas. Anal. Comput. Syst, Dec. 2020.

Proof of converse for $n = 4$

- Consider **any algorithm** independently implemented by 4 users

- Let Z be random time to first success

- First slot: Transmit with some prob p and observe $F[1]$:

$$\mathbb{E}\left[Z \mid F[1] = 0\right] \geq 1 + z_4^*$$
$$\mathbb{E}\left[Z \mid F[1] = 1\right] = 1$$
$$\mathbb{E}\left[Z \mid F[1] = 2\right] \geq ?? \quad \text{[Hard case: Groups } \{a, b\}, \{c, d\}\text{]}$$
$$\mathbb{E}\left[Z \mid F[1] = 3\right] \geq 2$$
$$\mathbb{E}\left[Z \mid F[1] = 4\right] \geq 1 + z_4^*$$

Proof idea: Emulation on a virtual system

- Pesky case of $\{a, b\}, \{c, d\}$.
- Want to bound expected remaining time under any algorithm for this pesky case:
$$\mathbb{E}[R] \geq 2$$
- Consider new system with **2 virtual users with enhanced capabilities:** Each user can send **any integer number** of packets per slot!
- Show virtual system has $\mathbb{E}[R_{virtual}] \geq 2$
- Show virtual system **can emulate** the $\{a, b\}, \{c, d\}$ case.

Why the problem is hard

- Indistinguishable users $\{1, 2, ..., n\}$.

- Feedback eventually lets us discern two groups:

$$\{k \text{ users}\}, \{n - k \text{ users}\}$$

- Should we throw one group away, or have first group transmit with prob p_1 and second with prob p_2?

- **Exponentially growing** (distributed) information state:
  1. User 1 history: $\{001101...\}$
  2. User 2 history: $\{110010...\}$
  3. User 3 history: $\{111001...\}$
  4. User 4 history: $\{111010...\}$

Matlab details

- Master Program:

  for $t \in \{1, \ldots, 100\}$:

  1. $X_1 = Player1DecisionAlg(t, Hist_1[t], Hist_2[t]);$
  2. $X_2 = Player2DecisionAlg(t, Hist_2[t], Hist_1[t]);$
  3. Update scores;
  4. Update history:

$$Hist_1[t] = [Hist_1[t]; X_1];$$
$$Hist_2[t] = [Hist_2[t]; X_2];$$

- Player subroutine:

$$\boxed{X = MyDecisionAlg(t, MyHistory[t], OpponentHistory[t]);}$$