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

<table>
<thead>
<tr>
<th></th>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example 100 slot Game

<table>
<thead>
<tr>
<th></th>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example 100 slot Game

<table>
<thead>
<tr>
<th></th>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td><strong>Collision</strong></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 100 slot Game

<table>
<thead>
<tr>
<th></th>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Collision</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 100 slot Game

<table>
<thead>
<tr>
<th></th>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td><strong>Collision</strong></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Example 100 slot Game**

<table>
<thead>
<tr>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 100 slot Game

<table>
<thead>
<tr>
<th></th>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Collision</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Idle</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 100 slot Game

<table>
<thead>
<tr>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 100 slot Game

<table>
<thead>
<tr>
<th>ALG 1</th>
<th>ALG 2</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Collision</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Idle</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALG 1 gets 1 point</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
<th>A9</th>
<th>A10</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A2</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>48.89</td>
<td>0</td>
<td>19.99</td>
<td>10.72</td>
<td>0</td>
<td>50.03</td>
<td>230.6444</td>
</tr>
<tr>
<td>A3</td>
<td>98.03</td>
<td>0</td>
<td>49.49</td>
<td>0</td>
<td>49.41</td>
<td>49.33</td>
<td>21.61</td>
<td>24.90</td>
<td>0</td>
<td>33.27</td>
<td>326.0706</td>
</tr>
<tr>
<td>A4</td>
<td>39.94</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>20.46</td>
<td>0</td>
<td>20.06</td>
<td>10.69</td>
<td>0</td>
<td>36.27</td>
<td>128.4562</td>
</tr>
<tr>
<td>A5</td>
<td>49.95</td>
<td>0</td>
<td>0.49</td>
<td>0</td>
<td>25.04</td>
<td>0</td>
<td>9.98</td>
<td>5.33</td>
<td>0</td>
<td>24.88</td>
<td>115.6895</td>
</tr>
<tr>
<td>A6</td>
<td>1</td>
<td>0</td>
<td>49.67</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>19.05</td>
<td>31.93</td>
<td>0</td>
<td>25.26</td>
<td>127.4265</td>
</tr>
<tr>
<td>A7</td>
<td>100</td>
<td>0</td>
<td>19.12</td>
<td>0.54</td>
<td>49.98</td>
<td>18.81</td>
<td>16.45</td>
<td>19.21</td>
<td>0.56</td>
<td>34.46</td>
<td>259.1658</td>
</tr>
<tr>
<td>A8</td>
<td>50.52</td>
<td>0</td>
<td>23.85</td>
<td>0.02</td>
<td>25.34</td>
<td>31.57</td>
<td>18.65</td>
<td>24.74</td>
<td>0.19</td>
<td>27.37</td>
<td>202.2952</td>
</tr>
<tr>
<td>A9</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>49.97</td>
<td>0</td>
<td>19.88</td>
<td>10.76</td>
<td>0</td>
<td>38.18</td>
<td>219.8058</td>
</tr>
<tr>
<td>A10</td>
<td>50.01</td>
<td>0</td>
<td>16.68</td>
<td>13.75</td>
<td>25.15</td>
<td>24.76</td>
<td>15.61</td>
<td>22.73</td>
<td>11.81</td>
<td>24.98</td>
<td>205.5207</td>
</tr>
</tbody>
</table>
Figures of merit for an algorithm

- **SelfCompetition score** $\alpha$: What is your expected score when playing an independent version of yourself?

- **NoCompetition score** $\beta$: What is your expected score when playing NeverTransmit?

- **HumanCompetition score** $\gamma$: 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 $t \in \{2, \ldots, 100\}$: $X[t] = X_{\text{opponent}}[t - 1]$

- **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 $\frac{1}{2}$ until either I score or the opponent scores.

  - If I score first, return to State 1.
  - If opponent scores first, go to State 4.

- **State 2:** Politely remain idle for one slot.

  - If opponent is idle, go to State 1.
  - If opponent transmits, go to State 3.

- **State 3:** Transmit repeatedly until I score.

  - If I score first, return to State 1.
  - If opponent transmits, go to State 3.

- **State 4:** Transmit repeatedly until collision.

  - If I score first, return to State 1.
  - If opponent is idle, go to State 2.
  - If opponent transmits, go to State 3.
Results of competition

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

<table>
<thead>
<tr>
<th></th>
<th>SelfComp $\alpha$</th>
<th>NoComp $\beta$</th>
<th>Tournament $\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-state</td>
<td>49.500</td>
<td>98.000</td>
<td>24.613</td>
</tr>
<tr>
<td>Tit-for-tat-0</td>
<td>0</td>
<td>0</td>
<td>20.410</td>
</tr>
<tr>
<td>Tit-for-tat-1</td>
<td>0</td>
<td>1</td>
<td>15.326</td>
</tr>
<tr>
<td>AlwaysTransmit</td>
<td>0</td>
<td>100</td>
<td>10.714</td>
</tr>
</tbody>
</table>

- 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

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

\[ T = 100 \implies \alpha \approx 49.500000000000000000000000000394 \]

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
     \textbf{Proof for } n = 3 \text{ agents; } n > 3 \text{ open}
  2. Nayyar and Teneketzis 2019
     \textbf{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
  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)
- $\ldots$
- $F[t] = n$ (Collision of $n$ users)

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

1. Measuring energy in collision
Proposed Alg for $n = 2$

Both users \textit{independently transmit with prob} $1/2$ every slot until first success.

- $Z =$ random time to first success.

- $z_2 = \mathbb{E}[Z]$  

- $z_2 = 2$
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 \( p \):

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

\[
\implies p^* = 0.411972
\]

\[
z_3 = 1.78795
\]
Proposed Alg for general \(n\)

Transmit with prob \(p\) and observe \(F[t]\):

- \(F[t] = 0\): Repeat
- \(F[t] = 1\): Done in 1
- \(F[t] = k \in \{2, \ldots, n - 2\}\):
  
  Choose better of groups: \(\{k\ \text{users}\}, \{n - k\ \text{users}\}\)

- \(F[t] = n - 1\): 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 \(n \in \{1, 2, 3, \ldots\}\)

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 $\mathbb{E}[Z_{\text{virtual}}] \geq z_4$

- Show virtual system can emulate actual system
  
  (so $\mathbb{E}[Z] \geq \mathbb{E}[Z_{\text{virtual}}]$)
Conclusions

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

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

2. NSF CCF-1718477
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 $F[1]$: 

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

▶ 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:

$$X = MyDecisionAlg(t, MyHistory[t], OpponentHistory[t]);$$