

# Trabalho: Análise Computacional de Jogos Estocásticos Cooperativos e Não Cooperativos

Disciplina: Aprendizado por Reforço e Controle  
Profs. Daniel Sadoc Menasché, Heudson Mirandola e Amit Bhaya

## Contexto

Este trabalho visa aprofundar o entendimento de **equilíbrios em jogos estocásticos** aplicados a sistemas de comunicação com múltiplos agentes — por exemplo, enlaces sem fio sujeitos a interferência, controle de potência e políticas de admissão de pacotes.

O notebook base (`team_game_vs_non_cooperative_simple_stochastic_game.py`) apresenta três modos principais de interação estratégica:

- a) **Jogo de Soma Zero (Zero-Sum)**
- b) **Jogo Não Cooperativo (Equilíbrio de Nash Descentralizado)**
- c) **Jogo de Time (Team Problem)**

Cada modo é resolvido por meio de **programação linear iterativa**, e os resultados são comparados em termos de throughput físico agregado, políticas de transmissão e políticas de admissão.

## Objetivos

- Compreender a formulação de jogos estocásticos envolvendo múltiplos agentes e restrições de potência e backlog.
- Explorar métodos alternativos de solução (Programação Linear versus **Problema de Complementaridade Linear – LCP**, resolvido via **algoritmo de Lemke**).
- Analisar efeitos paramétricos sobre o desempenho, estabilidade do buffer e comportamento do controlador de admissão.
- Identificar condições sob as quais o sistema opta por rejeitar pacotes, equilibrando throughput e estabilidade.

## Atividades Propostas

### Parte 1 – Revisão e entendimento do notebook

1. Execute o notebook fornecido e explique o papel das funções e células principais:

- Definição do modelo e dos parâmetros.
  - Solução dos três modos de jogo (Zero-Sum, Non-Coop, Coop).
  - Cálculo e visualização das políticas ótimas.
  - Avaliação do throughput físico agregado.
2. Descreva qualitativamente o significado das matrizes  $c_1$  e  $c_2$  e como elas mudam entre os modos.

## Parte 2 – Formulação e solução via algoritmo de Lemke (LCP)

1. O equilíbrio de Nash em jogos bilineares pode ser expresso como um **Problema de Complementaridade Linear (LCP)**:

$$\text{Encontrar } x \geq 0 \text{ tal que } Mx + q \geq 0, \quad x^\top(Mx + q) = 0$$

onde  $M$  é a matriz de interação e  $q$  contém os termos constantes das restrições.

- Derive  $M$  e  $q$  para a versão não cooperativa do jogo, considerando as distribuições ocupacionais  $(\rho_1, \rho_2)$ .
- Resolva o LCP utilizando o **algoritmo de Lemke** (por exemplo, via biblioteca `lemkelcp` ou implementação própria).
- Compare a solução obtida com a solução iterativa baseada em programação linear, mostrando numericamente que ambas produzem os mesmos equilíbrios.
- Documente as diferenças em tempo de execução, número de iterações e precisão.

## Parte 3 – Inclusão de restrição sobre o backlog médio

1. Acrescente uma restrição de **tamanho médio do buffer**, impondo:

$$\mathbb{E}[\text{Backlog}] \leq B_{\max}$$

onde  $B_{\max}$  é um valor limite escolhido (por exemplo, 3 ou 4 pacotes).

2. Mostre como essa restrição afeta o problema de otimização, podendo ser modelada como:

$$\sum_{s,b,a,d} b \cdot \rho(s, b, a, d) \leq B_{\max}$$

- Reexecute as simulações para diferentes valores de  $B_{\max}$  e observe:
  - Como as políticas de admissão mudam (em quais estados o sistema passa a rejeitar pacotes);
  - Como o throughput e o atraso médio se alteram.
- Inclua gráficos mostrando o **trade-off entre throughput e backlog**.
- Destaque cenários nos quais o controlador decide **não admitir pacotes**.

## Parte 4 – Análise paramétrica

1. Varie os parâmetros principais do sistema:
  - Probabilidade de chegada (`ArrProb`);
  - Limites de potência ( $v_1, v_2$ );
  - Número de estados de canal (`NLinkStates`);
  - Tamanho do buffer (`NBufferStates`).
2. Para cada variação:
  - Gere e plote as políticas ótimas de potência e admissão;
  - Recalcule o throughput físico e o backlog médio;
  - Compare com os modos cooperativo e não cooperativo.
3. Discuta:
  - Em que regimes o jogo cooperativo produz maior eficiência;
  - Quando a restrição de backlog força o sistema a rejeitar admissões;
  - O impacto da interferência e do custo de potência na estabilidade global.

## Parte 5 – Relatório Final

Monte um relatório contendo:

- Breve introdução teórica ao problema;
- Descrição da metodologia (LP, LCP/Lemke e restrição de backlog);
- Gráficos comparativos de políticas e desempenho;
- Discussão dos resultados e conclusões.

## Entrega

- Notebook executável (`.ipynb`) com comentários e gráficos;
- Relatório em PDF (máximo 6 páginas) resumindo os resultados.

## Critérios de Avaliação

Critério	Peso
Clareza e organização do código	15%
Utilização correta do LCP/Lemke	20%
Implementação da restrição de backlog médio	20%
Análise paramétrica e visualizações	20%
Discussão crítica e relatório final	25%

## **Extensões sugeridas – bonus extra**

- Investigar políticas com custo ponderado de atraso (penalizando backlog elevado);
- Explorar versões assimétricas (diferentes limites de potência ou probabilidades de chegada);
- Estender o modelo para 3 jogadores (múltiplos enlaces interferentes).