Jogos Gráficos

Renzo Gonzalo Gómez Diaz

Instituto de Matemática e Estatística
Universidade de São Paulo

Junho 2013

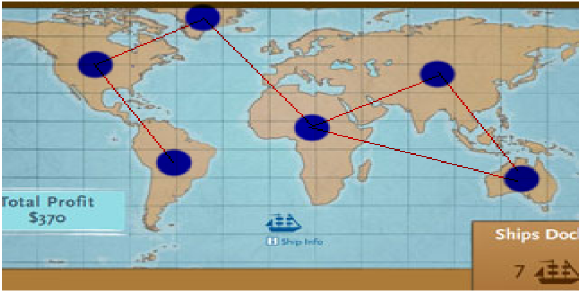
Introdução

- Para representar um jogo em forma normal, precisa-se uma quantidade exponencial de números que depende do número de jogadores e de estratégias.

- Jogos gráficos são uma representação que aproveita situações onde a utilidade de um jogador está determinada pelas ações de um subconjunto de jogadores.

Alguns exemplos:

- Balanceamento de carga numa rede de computadores.
- Intercâmbio comercial entre regiões (ou países).



Jogos Gráficos

Definição

Um jogo gráfico é um par $(G, \mathcal{M})$, onde G é um grafo não-dirigido, com conjunto de vértices $\{1, \ldots, n\}$, e $\mathcal{M}$ é o conjunto de n matrizes M_i, que descreve a utilidade do jogador i.

Jogos Gráficos

Definição

Um jogo gráfico é um par $(G, \mathcal{M})$, onde G é um grafo não-dirigido, com conjunto de vértices $\{1, \dots, n\}$, e $\mathcal{M}$ é o conjunto de n matrizes M_i, que descreve a utilidade do jogador i.

- Denotamos por S_i o conjunto de estratégias do jogador i. $S_i = \{0, 1\}$ para todo $i = 1, 2, \dots, n$.
- Neste caso, um perfil de estratégias mistas é dado por um vetor $\vec{p} \in [0, 1]^n$, tal que p_i é a probabilidade do jogador i escolher a estratégia 0.

Algoritmo TreeNash

- Precisa que o grafo definido pelo jogo seja uma árvore.
- Está composto de duas fases:
  1. Fase descendente: Calcula as melhores respostas de um vértice considerando as estratégias mistas da sua vizinhança.
  2. Fase ascendente: Devolve um equilíbrio de Nash.

Algoritmo TreeNash

- Precisa que o grafo definido pelo jogo seja uma árvore.
- Está composto de duas fases:
  1. Fase descendente: Calcula as melhores respostas de um vértice considerando as estratégias mistas da sua vizinhança.
  2. Fase ascendente: Devolve um equilíbrio de Nash.

Estrutura de dados

- Cada vértice V tem associado uma tabela binária $T(w, v)$ onde w é a estratégia mista do pai de V e v é a estratégia mista de V, no caso da raíz é $T(v)$.
- Cada entrada $T(w, v)$ tem uma lista de testemunhas (witness).

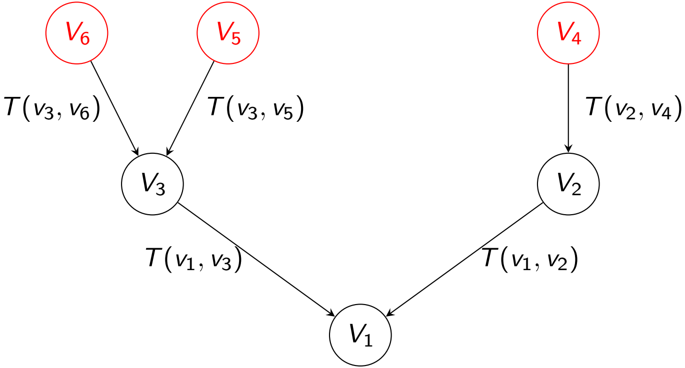
Algoritmo TreeNash - Início

1. Escolher um vértice Z arbitrariamente como raiz.
2. Encontrar a ordem dos vértices de G segundo uma busca em profundidade a partir de Z.
3. Para cada vértice $V \in V(G)$, fazemos $T(w, v) = 0$, $\forall v, w \in [0, 1]$.

Algoritmo TreeNash - Fase descendente

para todo $V \in V(G)$ na ordem dada pela DFS **faça**
    Seja W o pai do V (ou NIL se V é a raiz).
    se V é folha **então**
        para todo $w, v \in [0, 1]$, tal que $V = v$ é melhor resposta para $W = w$ **faça**
            $T(w, v) \leftarrow 1$
    senão
        Sejam $\vec{U} = (U_1, \ldots, U_k)$ os filhos de V; seja $T(v, u_i)$ a tabela passada por U_i para V.
        para todo $w, v \in [0, 1]$ **faça**
            para todo estratégia mista $\vec{u} = (u_1, \ldots, u_k)$ de $\vec{U}$ **faça**
                se $v = V$ é melhor resposta para $W = w, \vec{U} = \vec{u}$ **e** $T(v, u_i) = 1$ para $i = 1, \ldots, k$ **então**
                    $T(w, v) \leftarrow 1$
                    Adicione $\vec{u}$ à lista de testemunhas de $T(w, v)$
    Passar tabela $T(w, v)$ de V para W

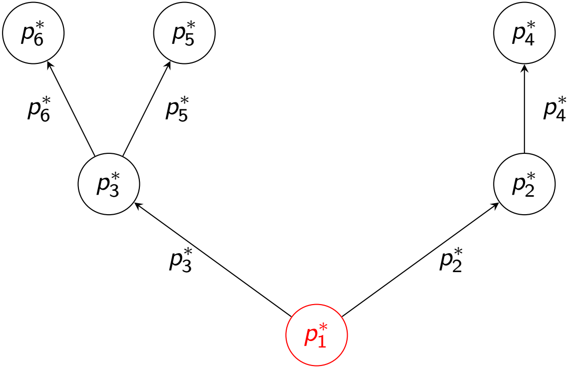
Algoritmo TreeNash - Fase descendente



Algoritmo TreeNash - Fase ascendente

para todo $V \in V(G)$ desde a raiz (em ordem reverso ao dado pela DFS) **faça**
    Seja $\vec{U} = (U_{\mathbf{1}}, \ldots, U_k)$ os filhos de V (ou vazio se V for uma folha)
    se V é raiz **então**
        Escolha arbitrariamente $v \in [0, 1]$, tal que $T(v) = 1$.
    senão
        Seja W o pai de V e (w, v) os valores passados por W a V.
    Etiquete V com o valor v
    Escolha qualquer testemunha $\vec{u}$ tal que $T(w, v) = 1$ (ou $T(v) = 1$)
    Para $i = 1, \ldots, k$ passe (v, u_i) de V para U_i

Algoritmo TreeNash - Fase ascendente



Notação:

- G^V : Subárvore de G com raiz em V.
- $\mathcal{M}^V_{W=w}$: Subconjunto de matrizes de $\mathcal{M}$, dos vértices em G^V, tal que a matriz M_V tem fixado a estratégia do seu pai $W = w$.

Notação:

- G^V : Subárvore de G com raiz em V.
- $\mathcal{M}^V_{W=w}$: Subconjunto de matrizes de $\mathcal{M}$, dos vértices em G^V, tal que a matriz M_V tem fixado a estratégia do seu pai $W = w$.

Proposição

Seja $V \in V(G)$ e W o pai de V. $T(w, v) = 1$ se existe um equilíbrio de Nash para $(G^V, \mathcal{M}^V_{W=w})$ tal que $V = v$.

Prova: Por indução na altura de G^V.
Se V é folha de G, então como $T(w, v) = 1$, pela condição da linha 4 da fase descendente, tem-se que $V = v$ é melhor resposta para $W = w$.
Logo, G^V está em equilíbrio se $V = v$ e $W = w$.

Se G^V tem altura maior do que 1. Sejam $\{U_1, U_2, \ldots, U_k\}$ os filhos de V em G e $\vec{u} = (u_1, \ldots, u_k)$ um testemunha de $T(w, v) = 1$. Então, $T(v, u_i) = 1$ para $i = 1, \ldots, k$, logo pela hipótese de indução, se V joga a estratégia mista v, existe um equilíbrio de Nash em $(G^{U_i}, \mathcal{M}^{U_i}_{V=v})$ tal que $U_i = u_i$ para $i = 1, \ldots, k$.
Alem disso, $V = v$ é melhor resposta para $W = w$ e $U_i = u_i$, logo $(G^V, \mathcal{M}^V_{W=w})$ tem um equilíbrio tal que $V = v$.

Corolário

Seja $(G, \mathcal{M})$ qualquer jogo gráfico tal que G é uma árvore. Então, o algoritmo TreeNash encontra um equilíbrio de Nash de $(G, \mathcal{M})$.

Corolário

Seja $(G, \mathcal{M})$ qualquer jogo gráfico tal que G é uma árvore. Então, o algoritmo TreeNash encontra um equilíbrio de Nash de $(G, \mathcal{M})$.

Problema: Precisa-se uma representação compacta para as tabelas $T(w, v)$ no algoritmo TreeNash.

ϵ-Equilíbrio de Nash

Um perfil de estratégias mistas $\vec{p}$ é um ϵ-equilíbrio de Nash se para todo jogador i e para todo $p'_i \in [0,1]$ tem-se que

$$M_i(\vec{p}) + \epsilon \quad \geq \quad M_i(p'_i, \vec{p}_{-i}).$$

ApproximateTreeNash

ϵ-Equilíbrio de Nash

Um perfil de estratégias mistas $\vec{p}$ é um ϵ-equilíbrio de Nash se para todo jogador i e para todo $p'_i \in [0,1]$ tem-se que

$$M_i(\vec{p}) + \epsilon \quad \geq \quad M_i(p'_i, \vec{p}_{-i}).$$

1. Dado um valor $\tau \in (0,1]$, cada jogador i pode escolher como estratégia mista $p_i \in \{0, \tau, 2\tau, \ldots, 1\}$.
2. Logo, no algoritmo TreeNash, cada vértice V passa uma tabela $T(w,v)$ com $(\frac{1}{\tau^2})$ entradas.

ApproximateTreeNash

Notamos que dado $\tau > 0$, o consumo de tempo do algoritmo está dominado pelas linhas 8-12 da fase descendente do TreeNash.

8 **para todo** $w, v \in [0,1]$ **faça**
9   **para todo** estratégia mista $\vec{u} = (u_1, \ldots, u_k)$ de $\vec{U}$ **faça**
10   **se** $v = V$ é melhor resposta para $W = w, \vec{U} = \vec{u}$ **e** $T(v, u_i) = 1$ para $i = 1, \ldots, k$ **então**
11             $T(w, v) \leftarrow 1$
12   Adicione $\vec{u}$ à lista de testemunhas de $T(w, v)$

ApproximateTreeNash

Notamos que dado $\tau > 0$, o consumo de tempo do algoritmo está dominado pelas linhas 8-12 da fase descendente do TreeNash.

```
8  para todo w, v ∈ [0, 1] faça
9      para todo estratégia mista u⃗ = (u_1, ..., u_k) de U⃗ faça
10         se v = V é melhor resposta para W = w, U⃗ = u⃗ e T(v, u_i) = 1 para i = 1, ..., k então
11             T(w, v) ← 1
12             Adicione u⃗ à lista de testemunhas de T(w, v)
```

Logo, aquele laço consome tempo $O((\frac{1}{\tau})^{2k})$.
Agora, vamos ver como a escolha de τ afeta a qualidade dos equilíbrios encontrados pelo algoritmo ApproximateTreeNash.

ApproximateTreeNash

Teorema [1]

Para todo $\epsilon > 0$, seja $\tau \leq \min(\epsilon/2^{k+2}(k \lg(k)), 2/k \lg^2(k/2))$. Então o algoritmo ApproximateTreeNash calcula um ϵ-equilíbrio de Nash para o jogo $(G, \mathcal{M})$.

Lema 1[1]

Seja $\vec{p}$,$\vec{q} \in [0,1]^k$ tal que $|p_i - q_i| \leq \tau$ para $i = 1, 2, \ldots, k$.
Se $\tau \leq 2/(k^2 \log(k/2))$,

$$|\prod_{i=1}^k p_i - \prod_{i=1}^k q_i| \quad \leq \quad (2k \lg(k))\tau.$$

Lema 2 [1]

Seja $\vec{p}$,$\vec{q}$ estratégias mistas, tal que $|p_i - q_i| \leq \tau$ para $i = 1, 2, \ldots, k$.
Se $\tau \leq 2/(k^2 \log(k/2))$,

$$|M_i(\vec{p}) - M_i(\vec{q})| \quad \leq \quad 2^{k+1}(k \lg(k))\tau.$$

Lema 3 [1]

Seja $\vec{p}$ um equilíbrio de Nash para $(G, \mathcal{M})$, e seja $\vec{q}$ o perfil mais próximo a $\vec{p}$ (em norma L_1) na τ-grade. Logo, se $\tau \leq 2/(k^2 \lg(k/2))$, $\vec{q}$ é um ϵ-equilíbrio de Nash, onde $\epsilon = 2^{k+2}(k \lg(k))\tau$.

No caso em que G é uma árvore pode-se mostrar o seguinte lema,

Lema 4 [1]

Sejam $\mathcal{J}$ um jogo de $k+2$ jogadores $(V, W, U_1, \ldots, U_k)$. Seja $M_V(v, \vec{u}, w)$ a utilidade esperada de V no perfil $(v, \vec{u}, w)$ e defina $\Delta(\vec{u}, w) = M_V(0, \vec{u}, w) - M_V(1, \vec{u}, w)$.
Sejam $I_1, \ldots, I_k \subseteq [0, 1]$ e

$$\mathcal{W} \quad = \quad \{w \in \ [0,1] : \exists \vec{u} \in I_1 \times \ldots \times I_k, \Delta(\vec{u}, w) = 0\}.$$

Então, $\mathcal{W}$ ou é vazio ou um intervalo em $[0, 1]$ ou a união de dois intervalos disjuntos em $[0, 1]$.

- Pelo Lema 4, fixado um intervalo para cada U_i, onde $T(v, u_i) = 1$, isto gera no máximo 2 interválos onde $T(w, v) = 1$.

- Logo, na fase descendente o número de regiões cresce de forma exponencial em cada nível.

- Pode-se mostrar que uma folha V tem no máximo 3 intervalos onde $T(w, v) = 1$. Portanto, o número de regiões é $O(3^n)$.

- Neste caso, o algoritmo é exponencial em n e não no grau máximo do grafo.

Referencias

Michael Kearns, Michael Littman, and Satinder Singh.
Graphical models for game theory.
In *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 253–260, San Francisco, CA, 2001. Morgan Kaufmann.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani.
Algorithmic Game Theory.
Cambridge University Press, New York, NY, USA, 2007.