Complexidade, Algoritmos de Aproximação e Teoria dos Jogos com aplicações em projetos de Redes Egoístas

Aluno: Alexandre Nobuo Kunieda

Orientador: Flávio Keidi Miyazawa

Instituto de Computação

Universidade Estadual de Campinas

1. Introdução

Problemas relacionados a teoria dos jogos vêm recebendo uma crescente atenção não apenas em matemática aplicada e economia, de onde originalmente surgiu, mas em diversas outras áreas do conhecimento onde esta é aplicada, como em biologia, filosofia, computação, lógica e etc. Especificamente em computação, pode-se citar sua aplicabilidade através do advento da internet e de sistemas distribuídos, situações nas quais o problema de redes egoístas em específico se encaixa.

Neste projeto, estivemos interessados no estudo de aspectos teóricos voltados a problemas de conectividade e Equilíbrio de Nash.

2. Teoria dos Jogos

Um jogo consiste de um conjunto de n jogadores, $\{1, 2, \ldots, n\}$. Durante um jogo, cada jogador i possui um conjunto S_i de escolhas possíveis que pode fazer, que chamamos de conjunto de estratégias.

Para jogar, cada jogador i escolhe uma estratégia $s_i \in S_i$. Dizemos que $s = (s_1, \ldots, s_n)$ é o *vetor de estratégias* escolhidas pelos jogadores e $S = \times_i S_i$ é o conjunto de todas maneiras possíveis que os jogadores podem escolher as estratégias.

O vetor de estratégias $s \in S$ selecionado pelos jogadores determina o resultado para cada jogador. Para especificar o jogo precisamos definir, para cada jogador, uma *ordem de preferência* sobre os resultados. Essa ordem de preferência deve ser uma relação binária completa, transitiva e reflexiva sobre o conjunto de todos vetores de estratégias S. Dado dois elementos de S, a relação para o jogador i diz qual dos dois resultados i prefere.

Uma maneira simples de representar isso é através de uma função benefício $u_i : S \to \Re$ para cada jogador i, que devolve um valor numérico para cada vetor de estratégias. Assim, se i mudou de estratégia, migrando de um vetor de estratégias s para um vetor s', então isso ocorreu somente porque $u_i(s) > u_i(s')$. Observe que a função de utilidade de um jogador está definida para cada $s \in S$ e portanto o benefício de um resultado para um jogador depende também das escolhas dos outros jogadores. Naturalmente, podemos também reescrever isto usando uma função de custo, em vez da função benefício. Dada uma função benefício u, podemos definir uma função de custo $c : S \to \Re$ fazendo $c_i(s) = -u_i(s)$. Neste caso, o jogador deseja minimizar seu custo.

A definição formal de estável pode variar com o tipo do jogo ou com que tipo de solução que estamos interessados. Para um vetor de estratégias $s \in S$, seja s_i a estratégia usada pelo jogador i e s_{-i} o vetor de estratégias de dimensão $n-1$ das estratégias jogadas por todos outros jogadores exceto o jogador i. Dada uma estratégia s'_i do jogador i, denotamos por (s'_i, s_{-i}) o vetor de estratégias substituindo a estratégia s_i por s'_i, e $u_i(s'_i, s_{-i})$ a função benefício do jogador i para a nova configuração.

Se um jogador tem uma estratégia ótima que é independente das estratégias escolhidas pelos outros jogadores, então este jogo possui uma *solução de estratégia dominante*. Mais formalmente, um vetor de estratégias $s \in S$ é uma *solução de estratégia dominante*, se para cada jogador i, e cada vetor alternativo de estratégias $s' \in S$, nós temos $u_i(s_i, s'_{-i}) \geq u_i(s'_i, s'_{-i})$.

O desejado é que todos jogos tenham uma solução estável obtida por estratégia dominante, mas a maioria dos jogos não possuem tal solução. Assim, buscamos uma definição de solução estável que seja um pouco menos restritiva e mais aplicável. Uma solução estável desejada é aquela na qual os jogadores agem de acordo com seus incentivos, escolhendo a melhor estratégia para si próprio dependendo das estratégias escolhidas pelos outros jogadores. Uma solução assim é chamada de *equilíbrio de Nash*.

Definição de equilíbrio de Nash: *Um vetor de estratégias $s \in S$ é dito estar em* equilíbrio de Nash *se, para todo jogador i e estratégia $s'_i \in S_i$, temos $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$.*

Isto é, um vetor de estratégias s está em equilíbrio de Nash se para todo jogador i a mudança da sua estratégia s_i para uma estratégia s'_i não melhora seu benefício (mantendo-se fixas as estratégias s_{-i} dos outros jogadores). Dentro um jogo, dizemos que um jogador i está *satisfeito* em relação a um vetor de estratégias s se $u(s'_i, s_{-i}) \leq u(s)$, para todo $s'_i \in S_i$. Caso contrário, diremos que i está *insatisfeito*. Claramente, em um equilíbrio de Nash, todos os jogadores estão satisfeitos.

3. Preço da Anarquia e Preço da Estabilidade

Para jogos em que temos vários equilíbrios de Nash, podemos ter diferentes benefícios em cada um dos equilíbrios. Algumas perguntas surgem: Qual a razão entre o custo do pior equilíbrio de Nash e a solução com benefício ótimo? Da mesma forma, qual a razão entre o melhor equilíbrio de Nash e a solução com benefício ótimo? Como medir a qualidade de um equilíbrio de Nash?

Uma abordagem é definir uma *função objetivo* em função de cada estado do jogo, que numericamente expressa o "custo social", ou seja, um custo arcado pelo sistema como um todo (não necessariamente refletindo os custos individuais dos jogadores). Dois tipos de função objetivo mais utilizadas são as funções *utilitária* e *equalitária*. A primeira é definida como a soma dos custos dos jogadores, a a segunda é definida como o custo máximo dos jogadores.

Existem duas principais maneiras para medir a ineficiência do equilíbrio: o *preço da anarquia* e o *preço da estabilidade*. O *preço da anarquia*, definido por Papadimitriou, é a medida mais popular, e resolve o problema da existência de múltiplos equilíbrios considerando a abordagem de pior caso. Mais especificamente, o preço da anarquia é definido como a razão entre o pior valor da função objetivo de um equilíbrio e o valor ótimo da função objetivo. Existem casos em que vários equilíbrios no sistema são razoavelmente eficientes, mas um caso extremo isolado faz com o o preço da anarquia seja grande. Para esses casos, usamos o *preço da estabilidade*, que é definido como a razão entre o melhor valor da função objetivo de um equilíbrio e o valor ótimo da função objetivo.

Preço da Anarquia:

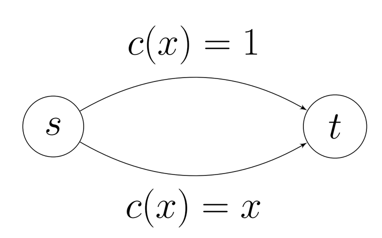
$$\max\left\{\frac{\max\{f(s) : s \in \varepsilon(J)\}}{\text{OPT}(J)} : \text{onde } j \text{ é uma instância do jogo}\right\}$$

Preço da Estabilidade:

$$\max\left\{\frac{\min\{f(s) : s \in \varepsilon(J)\}}{\text{OPT}(J)} : \text{onde } j \text{ é uma instância do jogo}\right\}$$

Onde consideramos um jogo J com conjunto de vetores de estratégias S, $f : S \to \Re$ a função de custo social, $\varepsilon(J)$ o conjunto de vetores de estratégias que estão em equilíbrio, e $\text{OPT}(J) = \min\{f(s) : s \in S\}$.

4. Exemplo de Pigou



Considere uma rede simples, de dois vértices s e t, e duas arestas disjuntas conectando o vértice s ao vértice t. Cada aresta contém uma função custo $c : \Re \to \Re$, que descreve o custo na aresta em função da quantidade de tráfego transmitido pela aresta.

A aresta superior tem custo constante, $c(x) = 1$, e a aresta inferior tem custo $c(x) = x$, ou seja, cresce lineamente à medida que se torna mais congestionada.

Suponha que haja uma unidade de tráfego, representando uma grande população de jogadores. No equilíbrio, único, todos os jogadores estarão na aresta inferior.

Assumimos que a função objetivo é minimizar o custo médio dos jogadores, e portanto o custo médio no equilíbrio é 1.

Esse, no entanto, não é o melhor custo total, que ocorre quando o tráfego é dividido igualmente entre as arestas, gerando custo $\frac{1}{2} + \frac{1}{2}\frac{1}{2} = \frac{3}{4}$.

O exemplo de Pigou tem a propriedade do preço da anarquia ser o mesmo do preço da estabilidade, isto é, $\frac{1}{\frac{3}{4}} = \frac{4}{3}$.

5. Nonatomic Selfish Routing

Este modelo é uma generalização natural do exemplo de Pigou (4) para redes mais complexas.

Um *selfish routing game* ocorre em uma rede, dada por um grafo $G = (V, E)$, um conjunto de k pares de vértices fonte-sorvedouro $(s_1, t_1), \ldots, (s_k, t_k)$, chamados *commodities*.

Cada jogador i está associado a um *commodity*, e escolhe algumas rotas para passar um tráfego $r_i \in \Re_+$ do vértice s_i ao t_i.

Seja P_i o conjunto de caminhos $s_i - t_i$ na rede. Temos $P_i \neq \emptyset \; \forall i$, e definimos $P = \bigcup_{i=1}^{k} P_i$.

Descrevemos as rotas escolhidas pelos jogadores usando um fluxo f, que é um vetor não-negativo indexado pelo conjunto P de caminhos fonte-sorvedouro.

Um fluxo f é viável se para um vetor r dos tráfegos de cada *commodity* se $\forall i \in \{1, 2, \ldots, k\}$, temos:

$$\sum_{p \in P_i} f_p = r_i$$

Finalmente, cada aresta e da rede tem uma função custo $c_e : \Re_+ \to \Re_+$ associada. Essa função é não-negativa, contínua, e não-decrescente.

Um *nonatomic selfish routing game* é definido pela tripla (G, r, c).

Definição de equilíbrio: Seja f um fluxo viável para uma instância do *nonatomic selfish routing game* (G, r, c). O fluxo está em equilíbrio se, para todo *commodity* $i \in \{1, 2, \ldots, k\}$ e cada par $p, p^* \in P_i$ dos caminhos $s_i - t_i$ com $f_p > 0$:

$$c_p(f) \leq c_{p^*}(f)$$

Definição de custo de um fluxo f:

$$C(f) = \sum_{p \in P} c_p(f)$$

$$C(f) = \sum_{e \in E[G]} c_e(f_e)$$

6. Atomic Selfish Routing

É definido se forma análoga ao *nonatomic selfish routing game*, ou seja, com o grafo $G = (V, E)$, k pares de fonte-sorvedouro (s_i, t_i), jogadores e tráfegos $r_i \in \Re_+$, e uma função $c_e : \Re_+ \to \Re_+$.

No entanto, um jogador i não pode dividir seu tráfego r_i em várias rotas, ou seja, o tráfego é indivisível, atÃ´mico.

Também descrevemos as rotas escolhidas pelos jogadores de forma um pouco diferente, de modo que o fluxo f agora é um vetor não-negativo indexado pelo conjunto P de caminhos fonte-sorvedouro e pelos jogadores. Assim $f_p(i)$ denota a quantidade de tráfego que o jogador i roteia pelo caminho p de s_i a t_i.

Dessa forma, um fluxo f é viável se $\forall i \in \{1, 2, \ldots, k\}$, $f_p(i) = r_i$ para exatamente um caminho p em P_i, e $f_p(i) = 0$ para todos os outros caminhos.

Definição de equilíbrio: Seja f um fluxo viável para uma instância do *atomic selfish routing game* (G, r, c). O fluxo está em equilíbrio se, para todo *commodity* $i \in \{1, 2, \ldots, k\}$ e cada par $p, p^* \in P_i$ dos caminhos $s_i - t_i$ com $f_p(i) > 0$:

$$c_p(f) \leq c_{p^*}(f^*)$$

Onde $f^* = f$, exceto que $f^*_p(i) = 0$ e $f^*_{p^*}(i) = r_i$.

Referências

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2001.

[2] Udi Manber, *Introduction to Algorithms: A Creative Approach*, Addison Wesley, 1989.

[3] Jayme Luiz Szwarcfiter, *Grafos e Algoritmos Computacionais*, Editora Campus, 1984.

[4] Flávio Keidi Miyazawa, *Introdução à Teoria dos Jogos Algorítmica*, 2010.

[5] Noam Nisan et al., *Algorithmic Game Theory*, Cambridge University Press, 2008.