MO758/MC758
Teoria dos Jogos Algorítmica

Rafael C. S. Schouery
rafael@ic.unicamp.br

Universidade Estadual de Campinas

1º semestre/2018

Leilões Combinatórios

Leilões

Leilões são uma parte importante da economia:

- Tanto do ponto de vista prático
- Como do ponto de vista teórico

Leilões existem desde a antiguidade e são usados para vender:

- Objetos de arte
- Commodities
- Transferir bens públicos para empresas privadas
- Direitos de utilização de recursos naturais
- etc...

Leilões de um único item

Existem várias formas de vender um único item

Leilões de carta fechada de primeiro preço:

- Cada comprador submete apenas um lance
- O comprador que der o maior lance ganha
- O vencedor paga o seu lance

Leilões de carta fechada de segundo preço:

- O vencedor paga o segundo maior lance

Leilões de um único item

Leilão inglês:

- O preço do item começa em um determinado valor
- Os participantes dão lances crescentes
- Termina quando ninguém aumenta o lance atual
  - Leilão de vela: termina quando acaba o tempo

O leilão inglês é parecido com o de segundo-preço

Leilão holandês:

- O preço do item começa em um determinado valor
- O preço diminui conforme o tempo passa
- O primeiro a manifestar interesse leva o item

O leilão holandês é parecido com o de primeiro-preço

Leilões Multi-unidade

Leilões multi-unidade:

- Queremos vender k itens idênticos
- Cada comprador informa um lance para cada quantidade de itens
- **Preços uniformes**: cada item tem o mesmo preço
- **Preços discriminatórios**: podemos vender os itens por preços diferentes

Leilões de bens digitais:

- Temos infinitas cópias do mesmo item
- Cada comprador deseja comprar uma cópia
- Preços uniformes ou discriminatórios

Leilões de demanda unitária

Leilões de demanda unitária:

- Queremos vender vários itens diferentes
- Mas cada comprador quer comprar apenas um item
- Cada comprador submete um lance para cada item
- Oferta única: Um item pode ser vendido apenas para um comprador
- Oferta limitada: Um item pode ser vendido um número de vezes
- Oferta ilimitada: Um item pode ser vendido inúmeras vezes

Leilões combinatórios

Em um leilão combinatório:

- Temos n compradores
- Queremos vender m itens

Valorações:

- para cada comprador i e cada conjunto S de itens, temos um valor $v_i(S)$

Restrições:

- Free-disposal: $v_i(S) \leq v_i(T)$ para todo $S \subseteq T$ e todo i
- Normalização: $v_i(\emptyset) = 0$ para todo i

As valorações são informações privadas

- Os compradores podem mentir ao relatar as valorações...

Leilões combinatórios

Alocação:

- Conjuntos $S_1, \ldots, S_n$ de itens tais que $S_i \cap S_j = \emptyset$ para todo $i \neq j$

Bem-estar social: $\sum_{i=1}^{n} v_i(S_i)$

Uma alocação é socialmente eficiente se ela maximiza o bem-estar social

Idealmente, queremos métodos eficientes e à prova de estratégia para maximizar o bem-estar social

Recap. de Complexidade Computacional

Como vimos anteriormente:

- Na década de 60, já existia a análise de algoritmos
- Vários problemas podiam ser resolvidos rapidamente
- Mas havia outros problemas para os quais não se conhecia algoritmos rápidos

Algoritmo rápido, ou eficiente:

- com complexidade de tempo $O(n^k)$ para alguma constante k (polinomial)

Recap. de Complexidade Computacional

Será que poderemos achar algoritmos rápidos para vários problemas práticos que pertencem a uma classe especial chamada NP?

Cook e Levin nos deram uma pista:

- Se existe algoritmo polinomial para o problema de Satifatibilidade Booleana (SAT), então existe algoritmo polinomial para qualquer problema em NP
- Este é o primeiro problema NP-Completo

Richard Karp mostrou que se existir um algoritmo polinomial para um problema (de uma lista de 21), então existe um algoritmo polinomial para o SAT

Recap. de Complexidade Computacional

Até hoje ninguém conseguiu encontrar um algoritmo polinomial para qualquer um dos problemas NP-completos

Conjectura: $\text{P} = \text{NP}$?

Muitas vezes problemas NP-completos são casos particulares ou podem ser reduzidos facilmente para outros de caráter mais prático, conhecidos como NP-difíceis

Algoritmos de Aproximação

Algoritmos de Aproximação:

- Executam em tempo polinomial
- Não encontram a solução ótima
- Mas encontram uma solução ”boa”

Para um problema P, uma instância I e um algoritmo A, denotamos por

- $\text{OPT}(I)$: o valor de uma solução ótima
- $A(I)$: o valor da solução encontrada por A

Um algoritmo A é uma α-aproximação se, para toda a instância I, temos que

- $A(I) \geq \alpha\,\text{OPT}(I)$ se o problema for de maximização
- $A(I) \leq \alpha\,\text{OPT}(I)$ se o problema for de minimização

Algoritmos de Aproximação

Um algoritmo A é uma α-aproximação se, para toda a instância I, temos que

- $A(I) \geq \alpha \, \mathsf{OPT}(I)$ se o problema for de maximização
- $A(I) \leq \alpha \, \mathsf{OPT}(I)$ se o problema for de minimização

Algoritmos de aproximação são uma forma de contornar a dificuldade de um problema NP-difícil, encontrando uma solução aproximada em tempo polinomial

Note a semelhança do conceito de razão de aproximação com a ideia de Preço da Anarquia, onde comparávamos a razão entre o pior equilíbrio e o ótimo social

Voltando para leilões combinatórios

- Temos n compradores
- Queremos vender m itens

Valorações:

- Para cada comprador i e cada conjunto S de itens, temos um valor $v_i(S)$

Restrições:

- *Free-disposal*: $v_i(S) \leq v_i(T)$ para todo $S \subseteq T$ e todo i
- Normalização: $v_i(\emptyset) = 0$ para todo i

Alocação:

- Conjuntos $S_1, \ldots, S_n$ de itens tais que $S_i \cap S_j = \emptyset$ para todo $i \neq j$

Bem-estar social: $\sum_{i=1}^{n} v_i(S_i)$

Caso de objetivo único (*single-minded*)

Cada participante está interessado em um único conjunto de itens

Cada valoração é dada por um par (S_i, v_i), representando $v_i(S) = v_i$ se $S \supseteq S_i$ e 0 caso contrário

Em suma, temos que decidir se o comprador i leva

- ou o conjunto S_i
  - se i recebe $S \supset S_i$, podemos parar de vender os itens em $S \setminus S_i$ para i
- ou o conjunto $\emptyset$
  - se i recebe $S \not\supseteq S_i$, podemos parar de vender os itens em S para i

Dificuldade de aproximar

Teorema: Seja m o número de itens, para todo $\varepsilon > 0$ não existe $(m^{1/2-\varepsilon})$-aproximação para a alocação ótima a menos que $\mathsf{P} = \mathsf{NP}$

Prova: Considere um grafo conexo $G = (V, E)$ e considere a seguinte instância do leilão:

- Cada aresta corresponde a um item
- Cada vértice corresponde a um comprador
- Um comprador i tem como valor (S_i, v_i) onde
  - S_i é o conjunto de arestas adjacentes a i em G
  - $v_i = 1$ para todo comprador i

Um conjunto $V' \subseteq V$ é um conjunto independente se não existe dois vértices em V' que são adjacentes

Existe um conjunto independente em G de tamanho k se e somente se existe uma alocação de valor k

Dificuldade de aproximar

Se existir uma $(m^{1/2-\varepsilon})$-aproximação para o problema da alocação ótima:

- podemos encontrar uma alocação de valor pelo menos $(m^{1/2-\varepsilon})opt$
- e encontrar um conjunto independente de tamanho pelo menos $(m^{1/2-\varepsilon})opt$
- como $m^{1/2-\varepsilon} \geq (n-1)^{1/2-\varepsilon} \geq n^{1/2-\varepsilon'}$
- esse conjunto independente tem tamanho pelo menos $n^{1/2-\varepsilon'}$

Isto é, temos uma $(n^{1/2-\varepsilon'})$-aproximação para o problema do conjunto independente máximo

Teorema[Håstad]: Não existe $n^{1/2-\varepsilon}$-aproximação para o problema do conjunto independente máximo a menos que P $=$ NP, onde n é o número de vértices do grafo

Aproximação à prova de estratégia

Guloso(S, v, n)

ordene v e S de modo que $\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}$
$W = \emptyset$
para $i = 1$ **até** n
    se $S_i \cap \bigcup_{k \in W} S_k = \emptyset$
        $W = W \cup \{i\}$
para $i \notin W$
    $p_i = 0$
para $i \in W$
    Seja j o menor índice tal que $S_j \cap S_i \neq \emptyset$, mas para todo $k < j$ em $W \setminus \{i\}$, $S_j \cap S_k = \emptyset$
    se j exisir
        $p_i = v_j / \sqrt{|S_j| / |S_i|}$
    senão
        $p_i = 0$
retorne W, p

Análise da Razão de Aproximação

Teorema: Seja A uma alocação ótima e seja W a alocação encontrada pelo algoritmo Guloso, então $\sum_{i \in A} v_i \leq \sqrt{m} \sum_{i \in W} v_i$

Prova: Para todo $i \in W$, seja

$$A_i = \{j \in A \colon j \geq i, S_i \cap S_j \neq \emptyset\}$$

Somando $v_j \leq (v_i/\sqrt{|S_i|})\sqrt{|S_j|}$ para $j \in A_i$, temos que

$$\sum_{j \in A_i} v_j \leq \frac{v_i}{\sqrt{|S_i|}} \sum_{j \in A_i} \sqrt{|S_j|}$$

Por Cauchy-Schwarz ($|u.v| \leq ||u||.||v||$),

$$\sum_{j \in A_i} \sqrt{|S_j|} \leq \sqrt{|A_i|} \sqrt{\sum_{j \in A_i} |S_j|}$$

Análise da Razão de Aproximação

$$\sum_{j \in A_i} v_j \leq \frac{v_i}{\sqrt{|S_i|}} \sum_{j \in A_i} \sqrt{|S_j|} \leq \frac{v_i}{\sqrt{|S_i|}} \sqrt{|A_i|} \sqrt{\sum_{j \in A_i} |S_j|}$$

Note que $|A_i| \leq |S_i|$ pois

- todo conjunto de A_i tem interseção não vazia com S_i
- A é uma alocação

Note também que $\sum_{j \in A_i} |S_j| \leq m$

- pois A é uma alocação

Assim,

$$\sum_{j \in A_i} v_j \leq \frac{v_i}{\sqrt{|S_i|}} \sqrt{|A_i|} \sqrt{\sum_{j \in A_i} |S_j|} \leq v_i \sqrt{m}$$

Análise da Razão de Aproximação

Para todo $i \in W$,

$$A_i = \{j \in A \colon j \geq i, S_i \cap S_j \neq \emptyset\}$$

Temos que

$$\sum_{j \in A_i} v_j \leq v_i \sqrt{m}$$

Note que $A \subseteq \bigcup_{i \in W} A_i$ e portanto,

$$\sum_{j \in A} v_j \leq \sum_{i \in W} \sum_{j \in A_i} v_j \leq \sqrt{m} \sum_{i \in W} v_i$$

À prova de estratégia

Lema: Um mecanismo para o leilão de objetivo único no qual perdedores pagam 0 é à prova de estratégia se e somente se satisfaz as seguintes condições:

(a) Monotonicidade: um participante que ganha ao declarar (S_i, v_i) continua ganhando se declarar (S_i', v_i') para todo $v_i' > v_i$ e todo $S_i' \subseteq S_i$

(b) Preço crítico: um vencedor paga o menor valor necessário para ganhar - o ínfimo de todos os valores v_i' tal que (S_i, v_i') ainda ganha

Teorema: Guloso é à prova de estratégia

Prova: É fácil ver que o Guloso satisfaz as duas condições

À prova de estratégia

Lema: Um mecanismo para o leilão de objetivo único no qual perdedores pagam 0 é à prova de estratégia se e somente se satisfaz as seguintes condições:

(a) Monotonicidade: um participante que ganha ao declarar (S_i, v_i) continua ganhando se declarar (S'_i, v'_i) para todo $v'_i > v_i$ e todo $S'_i \subseteq S_i$

(b) Preço crítico: um vencedor paga o menor valor necessário para ganhar - o ínfimo de todos os valores v_i^* tal que (S_i, v_i^*) ainda ganha

Prova: Tal mecanismo é individualmente racional pois

- perdedores pagam 0
- vencedores tem utilidade $v_i - v_i^* \geq 0$

Vamos mostrar que relatar (S'_i, v'_i) não é melhor do que relatar (S_i, v_i)

À prova de estratégia

Vamos mostrar que relatar (S_i', v_i') não é melhor do que relatar (S_i, v_i)

(S_i', v_i') não vale a pena

- Se não faz i ganhar
- ou S_i' não contém S_i

Vamos assumir que $S_i \subseteq S_i'$ e que (S_i', v_i') faz i ganhar

Vamos mostrar que

- (S_i', v_i') nunca é melhor do que (S_i, v_i')
- (S_i, v_i') nunca é melhor do que (S_i, v_i)

À prova de estratégia

(S'_i, v'_i) nunca é melhor do que (S_i, v'_i)

Como (S'_i, v'_i) ganha, por monotonicidade, (S_i, v'_i) também ganha

- p'_i o pagamento de i quando diz (S'_i, v'_i)
- p_i o pagamento de i quando diz (S_i, v'_i)

Como p_i é o preço crítico, (S_i, x) perde para todo $x < p_i$

Por monitonicidade, (S'_i, x) perde para todo $x < p_i$

Portanto, $p'_i \geq p_i$, isto é, a utilidade de i quando diz (S'_i, v'_i) não é maior do que quando diz (S_i, v'_i)

À prova de estratégia

(S_i, v_i') nunca é melhor do que (S_i, v_i)

Primeiro, suponha que (S_i, v_i) ganha

- $\tilde{p}_i$ o pagamento de i quando diz (S_i, v_i)

Se $v_i' \geq \tilde{p}_i$, então o pagamento de (S_i, v_i') é o mesmo do que (S_i, v_i)

Se $v_i' < \tilde{p}_i$, então (S_i, v_i') perde

Se (S_i, v_i) perde, então como (S_i, v_i') ganha,

- $v_i' \geq p_i > v_i$
- isto é, a utilidade é negativa

De volta ao caso geral

- Temos n compradores
- Queremos vender m itens

Valorações:

- Para cada comprador i e cada conjunto S de itens, temos um valor $v_i(S)$

Restrições:

- *Free-disposal*: $v_i(S) \leq v_i(T)$ para todo $S \subseteq T$ e todo i
- Normalização: $v_i(\emptyset) = 0$ para todo i

Alocação:

- Conjuntos $S_1, \ldots, S_n$ de itens tais que $S_i \cap S_j = \emptyset$ para todo $i \neq j$

Bem-estar social: $\sum_{i=1}^{n} v_i(S_i)$

Formulação linear inteira

Variáveis binárias: $x_{i,S} = 1$ se e somente se comprador i recebe o subconjunto S

$$
\begin{array}{llcll}
\text{maximize} & \displaystyle\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S) & & & \\
\text{sujeito a} & \displaystyle\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} & \leq & 1 & \forall j \in [m] \\
& \displaystyle\sum_{S \subseteq [m]} x_{i,S} & \leq & 1 & \forall i \in [n] \\
& x_{i,S} & \in & \{0, 1\} & \forall i \in [n], \forall S \subseteq [m]
\end{array}
$$

Relaxação linear inteira

Relaxamos a restrição de integralidade em $x_{i,S}$

$$
\begin{array}{lrcll}
\text{maximize} & \displaystyle\sum_{i=1}^{n} \sum_{S \subseteq [m]} x_{i,S} v_i(S) & & & \\
\text{sujeito a} & \displaystyle\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} & \leq & 1 & \forall j \in [m] \\
 & \displaystyle\sum_{S \subseteq [m]} x_{i,S} & \leq & 1 & \forall i \in [n] \\
 & x_{i,S} & \geq & 0 & \forall i \in [n], \forall S \subseteq [m]
\end{array}
$$

Como é o dual deste LP?

Dual

$$
\begin{array}{lrcll}
\text{minimize} & \displaystyle\sum_{i\in[n]} u_i + \sum_{j\in[m]} p_j & & & \\
\text{sujeito a} & \displaystyle u_i + \sum_{j\in S} p_j & \geq & v_i(S) & \forall i \in [n], \forall S \subseteq [m] \\
& u_i & \geq & 0 & \forall i \in [n] \\
& p_j & \geq & 0 & \forall j \in [m]
\end{array}
$$

Por folgas complementares, se $x_{i,S} > 0$, então:

- $u_i = v_i(S) - \sum_{j\in S} p_j$

Interpretação do dual:

- p_j é o preço do item j
- u_i é a utilidade do comprador i

Demanda

Dados preços $p_1, \ldots, p_m$ para os itens, uma demanda para o comprador i é um conjunto $S \subseteq [m]$ que maximize sua utilidade

A utilidade de i para S é $u_i(S) = v_i(S) - \sum_{j \in S} p_j$

Ou seja, uma demanda é um conjunto S tal que

$$u_i(S) \ = \ v_i(S) - \sum_{j \in S} p_j \ \geq \ v_i(S') - \sum_{j \in S'} p_j,$$

para qualquer $S' \subseteq [m]$

Equilíbrio Walrasiano

Um conjunto de preços não negativos $p_1^*, \ldots, p_m^*$ e uma alocação $S_1^*, \ldots, S_n^*$ são um equilíbrio Walrasiano se:

- para todo comprador i, S_i^* é uma demanda para i
- para todo item j não alocado, $p_j^* = 0$

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano então ele é economicamente eficiente (mesmo considerando alocações fracionárias)

Segundo Teorema do Bem-Estar Social: Se existe solução inteira ótima para o LP, então ela corresponde a um equilíbrio Walrasiano

Corolário: Um equilíbrio Walrasiano existe se e somente se o LP tem solução ótima inteira

Primeiro Teorema do Bem-Estar Social

Teorema: Se existe equilíbrio Walrasiano então ele é economicamente eficiente (mesmo considerando alocações fracionárias)

Prova: Cada comprador recebe sua demanda em um equilíbrio Walrasiano:

- $v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \geq v_i(S) - \sum_{j \in S} p_j^*$ para todo S

Toda solução X (fracionária) do LP satisfaz

- $\sum_{S \subseteq M} X_{i,S}^* \leq 1$ para todo comprador i

Portanto,

$$v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \geq \sum_{S \subseteq M} X_{i,S}^* \left(v_i(S) - \sum_{j \in S} p_j^* \right)$$

Primeiro Teorema do Bem-Estar Social

Somando para todos os compradores:

$$\sum_{i \in N} \left(v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \right) \geq \sum_{i \in N} \left(\sum_{S \subseteq M} X_{i,S}^* \left(v_i(S) - \sum_{j \in S} p_j^* \right) \right)$$

Portanto,

$$\begin{aligned} \sum_{i \in N} v_i(S_i^*) - \sum_{j \in M} p_j^* &\geq \sum_{i \in N} \sum_{S \subseteq M} X_{i,S}^* v_i(S) - \sum_{i \in N} \sum_{S \subseteq M} X_{i,S}^* \sum_{j \in S} p_j^* \\ &\geq \sum_{i \in N} \sum_{S \subseteq M} X_{i,S}^* v_i(S) - \sum_{j \in M} p_j^* \end{aligned}$$

pois $\sum_{i \in N} \sum_{S \subseteq [m] : j \in S} X_{i,S}^* \leq 1$

Segundo Teorema do Bem-Estar Social

Teorema: Se existe solução inteira ótima para o LP, então ela corresponde a um equilíbrio Walrasiano

Prova: Considere os conjuntos $S_1^*, S_2^*, \ldots, S_n^*$ dados por uma solução inteira ótima e variáveis p^* e u^* de uma solução dual ótima

Por folgas complementares, como $x_{i,S_i^*} > 0$, temos que

$$u_i^* = v_i(S_i^*) - \sum_{j \in S_i^*} p_j^*$$

I.e., para todo S, $v_i(S_i^*) - \sum_{j \in S_i^*} p_j^* \geq v_i(S) - \sum_{j \in S} p_j^*$

Também por folgas complementares, se $\sum_{i=1}^{n} \sum_{S \subseteq [m] : j \in S} x_{i,S} < 1$, então $p_j^* = 0$

Equilíbrio Walrasiano

Primeiro Teorema do Bem-Estar Social: Se existe equilíbrio Walrasiano então ele é economicamente eficiente (mesmo considerando alocações fracionárias)

Segundo Teorema do Bem-Estar Social: Se existe solução inteira ótima para o LP, então ela corresponde a um equilíbrio Walrasiano

Corolário: Um equilíbrio Walrasiano existe se e somente se o LP tem solução inteira

Equilíbrio Walrasiano

Exemplo onde não há equilíbrio Walrasiano:

Dois itens, a e b, e dois compradores, Ana e Beto, com as seguintes valorações

	$v(\{a\})$	$v(\{b\})$	$v(\{a,b\})$
Ana	2	2	2
Beto	0	0	3

O ótimo é dar $\{a, b\}$ para Beto e deixar Ana sem nada

Mas o conjunto vazio só é uma demanda para Ana se os preços de a e b forem pelo menos 2

Neste caso porém o preço de $\{a, b\}$ seria pelo menos 4 e $\{a, b\}$ não seria uma demanda para Beto

Consultas

Uma grande quantidade de informação dos compradores:

- $n \times 2^m$ números para representar os $v_i(S)$

Podemos pedir os números conforme o necessário, durante a execução do algoritmo

Consulta de valor: Dado um conjunto S, o comprador reporta o seu valor $v_i(S)$

Consulta de demanda: Dados preços para os itens, o comprador reporta uma demanda para esses preços

Consultas

Lema: Uma consulta de valor pode ser simulada por mt consultas de demanda, onde t é o número de bits de precisão nas representações das valorações

Lema: Pode ser necessário realizar um número exponencial de consultas de valor para simular uma única consulta de demanda

Teorema: A relaxação linear do programa inteiro para o problema da alocação máxima pode ser resolvida em tempo polinomial usando consultas de demanda

Complexidade de comunicação

Vimos que o problema de encontrar uma alocação ótima é NP-difícil

- mesmo no caso de objetivo único
- ou seja, mesmo quando temos que lidar com apenas n números e n conjuntos

Mas e se tivéssemos poder computacional ilimitado?

- Ainda assim, as valorações são representadas com $n2^m$ números
- Ou seja, os compradores precisam fornecer uma grande quantidade de números

Será que podemos reduzir a complexidade de comunicação?

- utilizando qualquer tipo de consulta...

O modelo de Yao

Dois jogadores, Ana e Beto, com valorações em $\{0, 1\}$

- isto é, $v_1, v_2 \colon 2^m \to \{0, 1\}$

Ana e Beto precisam se comunicar para encontrar uma alocação (S, S^c) que maximize $v_1(S) + v_2(S^c)$

Toda mensagem que Ana envia para Beto pode depender apenas de:

- mensagens anteriores enviadas por Beto
- sua valoração v_1

O equivalente vale para Beto

Teorema: Qualquer protocolo que encontra uma alocação ótima para todo par v_1 e v_2 utiliza pelo menos $\binom{m}{m/2}$ bits para a comunicação no pior caso

O modelo de Yao

Para uma valoração v, seja v^* a valoração dual:

- $v^*(S) = 1 - v(S^c)$ para todo $S \subseteq [m]$

Lema: Seja $v \neq u$ duas valorações. A sequência de bits transmitida para (v, v^*) não é igual a sequência de bits transmitida para (u, u^*)

Prova: Se for igual, então (v, v^*), (u, u^*), (v, u^*) e (u, v^*) produzem os mesmo conjuntos (S, S^c)

Como $u \neq v$, existe T tal que $v(T) \neq u(T)$

S.p.g., $v(T) = 1$ e $u(T) = 0$, isto é,
$v(S) + u^*(S^c) \geq v(T) + u^*(T^c) = 2$

Mas $v(S) + v^*(S^c) + u(S) + u^*(S^c) = 2$ e $u(S) + v^*(S^c) \leq 0$

Como $v(\emptyset) = 0$, temos que $v^*([m]) = 1$ e S não é ótimo para (u, v^*)

Complexidade de comunicação

Lema: Seja $v \neq u$ duas valorações. A sequência de bits transmitida para (v, v^*) não é igual a sequência de bits transmitida para (u, u^*)

Teorema: Qualquer protocolo que encontra uma alocação ótima para todo par v_1 e v_2 utiliza pelo menos $\binom{m}{m/2}$ bits para a comunicação no pior caso

Prova: Existe uma sequência de bits diferente para cada par (v, v^*), mas quantos pares diferentes existem?

Considere apenas valorações v tal que

- $v(S) = 0$ para todo $|S| < m/2$
- $v(S) = 1$ para todo $|S| > m/2$

Existem $2^{\binom{m}{m/2}}$ tais valorações e portanto existe uma comunicação com pelo menos $\binom{m}{m/2}$ bits

Simplificando o leilão combinatório

Até o momento, vimos que:

- O problema de encontrar a alocação ótima é NP-difícil
- O problema de encontrar a alocação ótima é dificil de aproximar por $m^{1/2-\varepsilon}$
  - ambos os resultados valhem mesmo para o caso de objetivo único
- A quantidade de números para representar as valorações é exponencial em m
- Qualquer tipo de consulta utilizada pode precisar de uma quantidade exponencial de bits

Em suma, leilões combinatórios são bem difíceis

Classes de Valorações

Porém, podemos restringir as valorações para deixar o problema mais tratável

Valorações subaditivas: para todo S e T, $v(S \cup T) \leq v(S) + v(T)$

Valorações submodulares: para todo $S \subseteq T$ e $x \notin T$, temos que $v(S \cup \{x\}) - v(S) \geq v(T \cup \{x\}) - v(T)$

Valorações com substitutos:

- se A é uma demanda para preços p
- e aumentarmos o preço dos itens,
- então existe uma demanda D que contém todos os itens de A que não subiram de preço

Leilões de um único item

Leilão inglês:

- O preço do item começa em um determinado valor
- Os participantes dão lances crescentes
- Termina quando ninguém aumenta o lance atual

O leilão inglês é um leilão iterativo

- em particular, é um leilão ascendente

Leilões ascendentes

Podemos pensar em leilões ascendentes para leilões combinatórios:

- O leiloeiro começa com os preços nulos (ou no mínimo)
- Através de consultas de demanda, o leiloeiro identifica quais são os itens desejados por um comprador com os preços atuais
- O leiloeiro aumenta alguns preços de algum modo, até decidir por uma alocação

Leilões ascendentes

Valoração v satisfaz a propriedade dos substitutos se:

- para todo par de preços dos itens $q \geq p$
- existe uma demanda com preços q
- que contém todos os itens da demanda com preço p que mantiveram seus preços

Formalmente: para toda demanda A para preços p, existe demanda D para preços q tal que $D \supseteq \{j \in A : p_j = q_j\}$

Apenas itens cujo preço subiu podem sair da demanda

Leilão de preços ascendentes de itens

PreçosAscendentesItens(m, n)

$p_j = 0$ **para** $j = 1$ **até** m
$S_i = \emptyset$ **para** $i = 1$ **até** n
enquanto existe i tal que S_i não é uma demanda de i para preços p_j se $j \in S_i$ e $p_j + \varepsilon$ se $j \notin S_i$
    Seja D_i uma tal demanda
    $p_j = p_j + \varepsilon$ para todo $j \in D_i \setminus S_i$
    $S_i = D_i$
    $S_k = S_k \setminus D_i$ para todo $k \neq i$
retorne $S_1, \ldots, S_n$

Equilíbrio ε-Walrasiano

Alocação $S_1, \dots, S_n$ e preços $p_1, \dots, p_m$ são equilíbrio ε-Walrasiano se

- todo item j com $p_j > 0$ pertence a algum S_i
- S_i para cada i é uma demanda de i para os preços p_j para $j \in S_i$ e $p_j + \varepsilon$ para $j \notin S_i$

Como os preços são ascendentes, o algoritmo termina em no máximo

$$m \times \frac{v_{\text{max}}}{\varepsilon}$$

passos, onde $v_{\text{max}} = \text{max}\{v_i(S) \colon i \in [n], S \subseteq [m]\}$

Teorema: Para valorações com substitutos, o algoritmo PreçosAscendentesItens produz um equilíbrio ε-Walrasiano

Assim a alocação produzida atinge um bem-estar social a εm do bem-estar social ótimo

Leilão de preços ascendentes de itens

Afirmação: A cada passo do algoritmo, para cada comprador i, $S_i \subseteq D_i$

Prova: É verdade no começo do algoritmo ($S_i = \emptyset$), então vamos focar quando um comprador i causou a mudança nos preços

Para i, fazemos $S_i = D_i$ e somamos ε nos preços dos itens em $D_i \setminus S_i$, portanto, a afirmação é valida

Para $k \neq i$, não é possível D_k perder um item e S_k não perder o item

- Se D_k perde um item, o preço aumentou e portanto o item foi para i
- Mas então removemos tal item de S_k

Leilão de preços ascendentes de itens

Teorema: Para valorações com substitutos, o algoritmo PreçosAscendentesItens produz um equilíbrio ε-Walrasiano

Prova:

- todo item com preço positivo pertence a algum S_i
- o algoritmo termina quando $S_i = D_i$

portanto, temos um equilíbrio ε-Walrasiano

Exemplo

Infelizmente, esse leilão não é à prova de estratégia

	$v(\{a\})$	$v(\{b\})$	$v(\{a,b\})$
Ana	4	4	4
Beto	5	5	10

Leilão deixa $\{a, b\}$ com Beto, ambos com preço ≈ 4

- Sua utilidade será de ≈ 2

Se Beto mente:

	$v(\{a\})$	$v(\{b\})$	$v(\{a,b\})$
Ana	4	4	4
Beto	5	0	5

Leilão deixa a com Beto e b com Ana, ambos com preço ≈ 0

- Com essa declaração, Beto fica com utilidade ≈ 5

Leilões ascendentes

Preços discriminatórios por pacotes de itens:

- Para cada i e cada $S \subseteq [m]$, um preço $p_i(S)$

Demanda para i: conjunto $D \in \arg\max_S \{v_i(S) - p_i(S)\}$

Equilíbrio competitivo: preços p e alocação $S = (S_1, \ldots, S_n)$ tq

- para todo i, conjunto S_i é uma demanda para i
- alocação maximiza o lucro do leiloeiro para p:

$$\sum_{i=1}^{n} p_i(S_i) \;\geq\; \sum_{i=1}^{n} p_i(T_i),$$

para qualquer outra alocação $T = (T_1, \ldots, T_n)$

Proposição: Em qualquer equilíbrio competitivo, a alocação maximiza o bem-estar social

Equilíbrio competitivo

Proposição: Em qualquer equilíbrio competitivo, a alocação maximiza o bem-estar social

Prova: Para todo comprador i, como S_i é uma demanda para i

$$v_i(S_i) - p_i(S_i) \geq v_i(T_i) - p_i(T_i)$$

Somando para todo i, temos que

$$\sum_i v_i(S_i) - \sum_i p_i(S_i) \geq \sum_i v_i(T_i) - \sum_i p_i(T_i)$$

Mas como o lucro é máximo,

$$\sum_i p_i(S_i) \geq \sum_i p_i(T_i)$$

e, portanto,

$$\sum_i v_i(S_i) \geq \sum_i v_i(T_i)$$

Leilão de preços ascendentes de pacotes

PreçosAscendentesPacotes(m, n)

$p_i(S) = 0$ **para** $i = 1$ **até** m e $S \subseteq [m]$

repita

  encontre alocação $T = (T_1, \ldots, T_n)$ que maximiza o lucro do leiloeiro com $p_i(T_i) > 0$ para todo $T_i \neq \emptyset$

  $L = \{i : T_i = \emptyset\}$

  para cada $i \in L$

    seja D_i uma demanda de i para os preços p_i

  retorne T, p **se** $D_i = \emptyset$ para todo i

  $p_i(D_i) = p_i(D_i) + \varepsilon$ **para cada** $i \in L$ tal que $D_i \neq \emptyset$

Equilíbrio ε-competitivo

Um conjunto S é uma ε-demanda para i se
$v_i(S) - p_i(S) \geq v_i(T) - p_i(T) - \varepsilon$ para todo $T \subseteq [m]$

Equilíbrio ε-competitivo: preços p e alocação $S = (S_1, \ldots, S_n)$ tal que

- para todo i, conjunto S_i é uma ε-demanda para i
- alocação maximiza o lucro do leiloeiro para p:

$$\sum_{i=1}^{n} p_i(S_i) \;\geq\; \sum_{i=1}^{n} p_i(T_i),$$

para qualquer outra alocação $T = (T_1, \ldots, T_n)$

Teorema: O leilão de preços ascendentes para pacotes termina com um equilíbrio ε-competitivo

Leilão de preços ascendentes para pacotes

Teorema: O leilão de preços ascendentes para pacotes termina com um equilíbrio ε-competitivo

Prova: Em primeiro lugar, note que o $p_i(S_i) \leq v_i(S_i) + \varepsilon$ e portanto o algoritmo eventualmente termina (mas pode demorar um tempo exponencial)

Além disso, a cada passo escolhemos uma alocação que maximiza o lucro do leiloeiro

Ou seja, basta provar que cada comprador recebe uma ε-demanda

Leilão de preços ascendentes para pacotes

Compradores que não recebe item, claramente recebem suas demandas (condição de término do algoritmo)

O algoritmo mantém a propriedade que todo pacote T_i com preço $p_i(T_i) > 0$ é uma ε-demanda

Considere a última vez que T_i foi uma demanda

- Ele se tornou uma ε-demanda imediatamente depois
- Até o final do algoritmo, mudamos apenas o preço de outros pacotes

Como o algoritmo apenas aloca conjuntos não-vazios com preços positivos, o resultado segue

Leilão de preços ascendentes para pacotes

Teorema: O leilão de preços ascendentes para pacotes termina com um equilíbrio ε-competitivo

Em particular, temos uma solução a uma distância de $n\varepsilon$ da ótima

O algoritmo pode consumir tempo exponencial

- mas funciona para qualquer tipo de valoração