Leilões combinatórios

n participantes m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Leilões combinatórios

n participantes m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

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.

Leilões combinatórios

n participantes $\qquad$ m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

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.

Leilão combinatório $\times$ vários leilões de um único item.

Leilões combinatórios

n participantes $\qquad$ m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

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.

Leilão combinatório $\times$ vários leilões de um único item.

- S e T se complementam se $v_i(S) + v_i(T) < v_i(S \cup T)$.
- S e T se substituem se $v_i(S) + v_i(T) > v_i(S \cup T)$.

Alocações

n participantes m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocações

n participantes $\qquad$ m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Alocações

n participantes $\quad\quad m$ itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

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

Alocações

n participantes $\qquad$ m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

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

Alocação socialmente eficiente:
maximiza o bem-estar social.

Alocações

n participantes $\qquad$ m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

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

Alocação socialmente eficiente:
maximiza o bem-estar social.

Valorações são informação privada.

Queremos métodos eficientes para
maximizar o bem-estar social.

Dificuldades

n participantes $\qquad$ m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

Dificuldades

n participantes m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

- Mesmo para casos particulares, encontrar alocações ótimas pode ser um problema difícil (NP-difícil).

Dificuldades

n participantes m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

- Mesmo para casos particulares, encontrar alocações ótimas pode ser um problema difícil (NP-difícil).
- Valorações tem tamanho exponencial em m.
Como lidar com isso?

Dificuldades

n participantes $\qquad$ m itens

valorações: para cada $i \in [n]$,
um valor $v_i(S)$ para cada $S \subseteq [m]$.

Alocação: conjuntos $S_1, \ldots, S_n$ de itens tq
$S_i \cap S_j = \emptyset$ para todo $i \neq j$.

- Mesmo para casos particulares, encontrar alocações ótimas pode ser um problema difícil (NP-difícil).
- Valorações tem tamanho exponencial em m.
  Como lidar com isso?
- Comportamento estratégico: como garantir que os participantes declararão suas reais valorações?

Caso de objetivo único

Caso particular: *single-minded*

Cada participante está interessado em um único conjunto.

Caso de objetivo único

Caso particular: *single-minded*
Cada participante está interessado em um único conjunto.

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.

Caso de objetivo único

Caso particular: *single-minded*
Cada participante está interessado em um único conjunto.

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.

Complexidade computacional:
encontrar uma alocação ótima é NP-difícil.

Caso de objetivo único

Caso particular: *single-minded*
Cada participante está interessado em um único conjunto.

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.

Complexidade computacional:
encontrar uma alocação ótima é NP-difícil.

Conjunto independente: Dado um grafo G e um inteiro k,
determinar se G tem um conjunto independente de
tamanho k.

Caso de objetivo único

Caso particular: *single-minded*
Cada participante está interessado em um único conjunto.

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.

Complexidade computacional:
encontrar uma alocação ótima é NP-difícil.

Conjunto independente: Dado um grafo G e um inteiro k, determinar se G tem um conjunto independente de tamanho k.

Redução do problema do conjunto independente para a versão de decisão do problema da alocação ótima.

Complexidade computacional

Teorema: O problema da alocação ótima é NP-difícil.

É pior que isso: a redução preserva aproximação, assim o problema herda a dificuldade do conjunto independente.

Complexidade computacional

Teorema: O problema da alocação ótima é NP-difícil.

É pior que isso: a redução preserva aproximação, assim o problema herda a dificuldade do conjunto independente.

Teorema: Para todo $\epsilon > 0$, não existe $n^{1-\epsilon}$-aproximação para o conjunto independente a menos que P $=$ NP, onde n é o número de vértices do grafo.

Complexidade computacional

Teorema: O problema da alocação ótima é NP-difícil.

É pior que isso: a redução preserva aproximação, assim o problema herda a dificuldade do conjunto independente.

Teorema: Para todo $\epsilon > 0$, não existe $n^{1-\epsilon}$-aproximação para o conjunto independente a menos que P $=$ NP, onde n é o número de vértices do grafo.

Como o número de arestas de um grafo é no máximo n^2, e as arestas são os itens na redução, vale o seguinte:

Complexidade computacional

Teorema: O problema da alocação ótima é NP-difícil.

É pior que isso: a redução preserva aproximação, assim o problema herda a dificuldade do conjunto independente.

Teorema: Para todo $\epsilon > 0$, não existe $n^{1-\epsilon}$-aproximação para o conjunto independente a menos que P $=$ NP, onde n é o número de vértices do grafo.

Como o número de arestas de um grafo é no máximo n^2, e as arestas são os itens na redução, vale o seguinte:

Teorema: Para todo $\epsilon > 0$, não existe $m^{1/2-\epsilon}$-aproximação para a alocação ótima a menos que P $=$ NP, onde m é o número de itens.

Aproximação à prova de estratégia

Um algoritmo guloso...

Aproximação à prova de estratégia

Um algoritmo guloso...

GULOSO (S, v, n)

ordene os participantes de modo que

$$\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}.$$

$W \leftarrow \emptyset$

para $i \leftarrow 1$ **até** n **faça**

  se $S_i \cap \cup_{k \in W} S_k = \emptyset$

    então $W \leftarrow W \cup \{i\}$

devolva W

Aproximação à prova de estratégia

Um algoritmo guloso...

GULOSO (S, v, n)

ordene os participantes de modo que
$$\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}.$$
$W \leftarrow \emptyset$
para $i \leftarrow 1$ **até** n **faça**
  se $S_i \cap \cup_{k \in W} S_k = \emptyset$
    então $W \leftarrow W \cup \{i\}$
devolva W

Mas e os preços?

Aproximação à prova de estratégia

Um algoritmo guloso...

GULOSO (S, v, n)
1 ordene os participantes de modo que
$$\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}.$$
2 $W \leftarrow \emptyset$
3 **para** $i \leftarrow 1$ **até** n **faça**
4 **se** $S_i \cap \cup_{k \in W} S_k = \emptyset$
5 **então** $W \leftarrow W \cup \{i\}$
6 **devolva** W

Mas e os preços?

Infelizmente preços VCG podem não funcionar dado que a alocação prduzida não é ótima.

Aproximação à prova de estratégia

GULOSO (S, v, n)

ordene v e S de modo que $\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}$.
$W \leftarrow \emptyset$
para $i \leftarrow 1$ **até** n **faça**
    se $S_i \cap \cup_{k \in W} S_k = \emptyset$
        então $p_i \leftarrow$ PREÇOCRÍTICO(S, v, n, i, W)
            $W \leftarrow W \cup \{i\}$
        senão $p_i \leftarrow 0$
devolva W, p

Aproximação à prova de estratégia

GULOSO (S, v, n)

ordene v e S de modo que $\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}$.
$W \leftarrow \emptyset$
para $i \leftarrow 1$ **até** n **faça**
  se $S_i \cap \cup_{k \in W} S_k = \emptyset$
    então $p_i \leftarrow$ PREÇOCRÍTICO(S, v, n, i, W)
      $W \leftarrow W \cup \{i\}$
    senão $p_i \leftarrow 0$
devolva W, p

PREÇOCRÍTICO (S, v, n, i, W)

para $j \leftarrow i + 1$ **até** n **faça**
  se $S_j \cap \cup_{k \in W} S_k = \emptyset$
    então se $S_j \cap S_i \neq \emptyset$
      então devolva $\frac{v_j}{\sqrt{|S_j|}} \sqrt{|S_i|}$
      $W \leftarrow W \cup \{j\}$
devolva 0

Aproximação à prova de estratégia

GULOSO (S, v, n)

ordene v e S de modo que $\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}$.
$W \leftarrow \emptyset$
para $i \leftarrow 1$ **até** n **faça**
  se $S_i \cap \cup_{k \in W} S_k = \emptyset$
    então $p_i \leftarrow$ PREÇOCRÍTICO(S, v, n, i, W)
      $W \leftarrow W \cup \{i\}$
    senão $p_i \leftarrow 0$
devolva W, p

Preço p_i: valor limite que faz i deixar de ganhar o leilão.

Aproximação à prova de estratégia

GULOSO (S, v, n)

ordene v e S de modo que $\frac{v_1}{\sqrt{|S_1|}} \geq \cdots \geq \frac{v_n}{\sqrt{|S_n|}}$.

$W \leftarrow \emptyset$

para $i \leftarrow 1$ **até** n **faça**

se $S_i \cap \cup_{k \in W} S_k = \emptyset$

então $p_i \leftarrow$ PREÇOCRÍTICO(S, v, n, i, W)

$W \leftarrow W \cup \{i\}$

senão $p_i \leftarrow 0$

devolva W, p

Preço p_i: valor limite que faz i deixar de ganhar o leilão.

Algoritmo obviamente polinomial.

Análise

Resta mostrar que

- é uma $\sqrt{m}$-aproximação;
- é a prova de estratégia.

Análise

Resta mostrar que

- é uma $\sqrt{m}$-aproximação;
- é a prova de estratégia.

Teorema: GULOSO é uma $\sqrt{m}$-aproximação.

Prova feita na aula.

Análise

Resta mostrar que

- é uma $\sqrt{m}$-aproximação;
- é a prova de estratégia.

Teorema: GULOSO é uma $\sqrt{m}$-aproximação.

Prova feita na aula.

Teorema: GULOSO é à prova de estratégia.

Prova feita na aula.