Combinatória Poliédrica
Otimização Combinatória e PLI

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

Universidade Estadual de Campinas

Atualizado em: 2023-09-27 15:58

Otimização Combinatória

Em um problema de otimização combinatória:

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c
- Queremos encontrar $S \in \mathcal{S}$

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c
- Queremos encontrar $S \in \mathcal{S}$
- Que maximiza/minimiza $c(S)$

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c
- Queremos encontrar $S \in \mathcal{S}$
- Que maximiza/minimiza $c(S)$

Em princípio, eu poderia analisar cada $S \in \mathcal{S}$

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c
- Queremos encontrar $S \in \mathcal{S}$
- Que maximiza/minimiza $c(S)$

Em princípio, eu poderia analisar cada $S \in \mathcal{S}$

- Mas, em geral, $\mathcal{S}$ é dado implicitamente por uma entrada

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c
- Queremos encontrar $S \in \mathcal{S}$
- Que maximiza/minimiza $c(S)$

Em princípio, eu poderia analisar cada $S \in \mathcal{S}$

- Mas, em geral, $\mathcal{S}$ é dado implicitamente por uma entrada
- E tem tamanho exponencial no tamanho da entrada

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c
- Queremos encontrar $S \in \mathcal{S}$
- Que maximiza/minimiza $c(S)$

Em princípio, eu poderia analisar cada $S \in \mathcal{S}$

- Mas, em geral, $\mathcal{S}$ é dado implicitamente por uma entrada
- E tem tamanho exponencial no tamanho da entrada

Na maioria das vezes, definir se uma solução é viável é fácil

Otimização Combinatória

Em um problema de otimização combinatória:

- Temos um conjunto finito $\mathcal{S}$ de soluções viáveis
- E uma função objetivo c
- Queremos encontrar $S \in \mathcal{S}$
- Que maximiza/minimiza $c(S)$

Em princípio, eu poderia analisar cada $S \in \mathcal{S}$

- Mas, em geral, $\mathcal{S}$ é dado implicitamente por uma entrada
- E tem tamanho exponencial no tamanho da entrada

Na maioria das vezes, definir se uma solução é viável é fácil

- O difícil é achar a solução ótima

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

- E é um conjunto finito de elementos

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

- E é um conjunto finito de elementos
- $c\colon E \to \mathbb{R}$ é uma função de custo

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

- E é um conjunto finito de elementos
- $c \colon E \to \mathbb{R}$ é uma função de custo
- $\mathcal{S} \subseteq 2^E$ é um conjunto de soluções viáveis

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

- E é um conjunto finito de elementos
- $c\colon E \to \mathbb{R}$ é uma função de custo
- $\mathcal{S} \subseteq 2^E$ é um conjunto de soluções viáveis

E o objetivo é encontrar $S \in \mathcal{S}$ que maximiza $c(S) = \sum_{e \in S} c(e)$

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

- E é um conjunto finito de elementos
- $c\colon E \to \mathbb{R}$ é uma função de custo
- $\mathcal{S} \subseteq 2^E$ é um conjunto de soluções viáveis

E o objetivo é encontrar $S \in \mathcal{S}$ que maximiza $c(S) = \sum_{e \in S} c(e)$

Isto é, queremos encontrar um subconjunto de E que maximize a soma dos custos de seus elementos

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

- E é um conjunto finito de elementos
- $c\colon E \to \mathbb{R}$ é uma função de custo
- $\mathcal{S} \subseteq 2^E$ é um conjunto de soluções viáveis

E o objetivo é encontrar $S \in \mathcal{S}$ que maximiza $c(S) = \sum_{e \in S} c(e)$

Isto é, queremos encontrar um subconjunto de E que maximize a soma dos custos de seus elementos

- Portanto, nossa função objetivo é linear

Otimização Combinatória Linear

Vamos focar uma classe particular de problemas de otimização combinatória

Temos uma tripla da forma $(E, c, \mathcal{S})$:

- E é um conjunto finito de elementos
- $c\colon E \to \mathbb{R}$ é uma função de custo
- $\mathcal{S} \subseteq 2^E$ é um conjunto de soluções viáveis

E o objetivo é encontrar $S \in \mathcal{S}$ que maximiza $c(S) = \sum_{e \in S} c(e)$

Isto é, queremos encontrar um subconjunto de E que maximize a soma dos custos de seus elementos

- Portanto, nossa função objetivo é linear

Apesar de mais restrito, muitos problemas de interesse são lineares

Exemplos de Problemas

Emparelhamento Máximo: Dado um grafo $G = (V, A)$ e uma função de custo $c\colon A \to \mathbb{R}$, achar um emparalhemento F em G tal que $c(F)$ seja máximo.

Exemplos de Problemas

Emparelhamento Máximo: Dado um grafo $G = (V, A)$ e uma função de custo $c\colon A \to \mathbb{R}$, achar um emparalhemento F em G tal que $c(F)$ seja máximo.

Caminho Mínimo: Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V e um função de custo $c\colon A \to \mathbb{R}$, achar um caminho P de s a t em D tal que $c(P)$ seja mínimo.

Exemplos de Problemas

Emparelhamento Máximo: Dado um grafo $G = (V, A)$ e uma função de custo $c\colon A \to \mathbb{R}$, achar um emparalhemento F em G tal que $c(F)$ seja máximo.

Caminho Mínimo: Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V e um função de custo $c\colon A \to \mathbb{R}$, achar um caminho P de s a t em D tal que $c(P)$ seja mínimo.

(s, t)-**Corte Mínimo**: Dado um grafo orientado $D = (V, A)$, dois vértices distintos s, t em V e um função de capacidade $c\colon A \to \mathbb{R}$, encontrar um (s, t)-corte S tal que $c(S)$ seja mínimo.

Exemplos

Árvore Geradora Mínima: Dado um grafo $G = (V, A)$ e uma função custo $c\colon A \to \mathbb{R}$, achar uma árvore geradora F em G tal que $c(F)$ seja mínimo.

Exemplos

Árvore Geradora Mínima: Dado um grafo $G = (V, A)$ e uma função custo $c\colon A \to \mathbb{R}$, achar uma árvore geradora F em G tal que $c(F)$ seja mínimo.

Caixeiro Viajante: Dado um grafo $G = (V, A)$ e uma função de custo $c\colon A \to \mathbb{R}$, encontrar um circuito hamiltoniano C em G tal que $c(C)$ seja mínimo.

Exemplos

Árvore Geradora Mínima: Dado um grafo $G = (V, A)$ e uma função custo $c\colon A \to \mathbb{R}$, achar uma árvore geradora F em G tal que $c(F)$ seja mínimo.

Caixeiro Viajante: Dado um grafo $G = (V, A)$ e uma função de custo $c\colon A \to \mathbb{R}$, encontrar um circuito hamiltoniano C em G tal que $c(C)$ seja mínimo.

Steiner em Grafos: Dado um grafo $G = (V, A)$, um conjunto de vértices $Z \subseteq V$ chamados terminais e uma função de custo $c\colon A \to \mathbb{R}$, encontrar $S \subseteq A$ tal que $G[S]$ conecta todos os terminais e $c(S)$ seja mínimo.

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Seja $\mathbb{R}^E$

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Seja $\mathbb{R}^E$

- o espaço vetorial real $|E|$-dimensional

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Seja $\mathbb{R}^E$

- o espaço vetorial real $|E|$-dimensional
- em que cada $\boldsymbol{x} \in \mathbb{R}^E$ tem seus componentes indexados por E

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Seja $\mathbb{R}^E$

- o espaço vetorial real $|E|$-dimensional
- em que cada $\boldsymbol{x} \in \mathbb{R}^E$ tem seus componentes indexados por E
  - Isto é, $\boldsymbol{x} = (x_e)_{e \in E}$

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Seja $\mathbb{R}^E$

- o espaço vetorial real $|E|$-dimensional
- em que cada $\boldsymbol{x} \in \mathbb{R}^E$ tem seus componentes indexados por E
  - Isto é, $\boldsymbol{x} = (x_e)_{e \in E}$

Para $S \in \mathcal{S}$, associamos S a um $\chi^S \in \mathbb{R}^E$ da seguinte forma:

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Seja $\mathbb{R}^E$

- o espaço vetorial real $|E|$-dimensional
- em que cada $\boldsymbol{x} \in \mathbb{R}^E$ tem seus componentes indexados por E
  - Isto é, $\boldsymbol{x} = (x_e)_{e \in E}$

Para $S \in \mathcal{S}$, associamos S a um $\chi^S \in \mathbb{R}^E$ da seguinte forma:

$$\chi_e^S = \begin{cases} 1, & \text{se } e \in S \\ 0, & \text{caso contrário} \end{cases}$$

Vetor de Incidência

Digamos que temos uma tripla $(E, c, \mathcal{S})$

Seja $\mathbb{R}^E$

- o espaço vetorial real $|E|$-dimensional
- em que cada $\boldsymbol{x} \in \mathbb{R}^E$ tem seus componentes indexados por E
  - Isto é, $\boldsymbol{x} = (x_e)_{e \in E}$

Para $S \in \mathcal{S}$, associamos S a um $\chi^S \in \mathbb{R}^E$ da seguinte forma:

$$\chi_e^S = \begin{cases} 1, & \text{se } e \in S \\ 0, & \text{caso contrário} \end{cases}$$

$\boldsymbol{\chi}^S$ é o vetor de incidência de S

Politopo

Consideramos o politopo $P \subseteq \mathbb{R}^E$ definido como:

$$P = \operatorname{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\}$$

Politopo

Consideramos o politopo $P \subseteq \mathbb{R}^E$ definido como:

$$P = \mathrm{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\}$$

Exercício. Seja P o politopo definido acima. Mostre que o conjunto dos vértices de P é precisamente o conjunto dos vetores de incidência dos elementos de $\mathcal{S}$.

Politopo

Consideramos o politopo $P \subseteq \mathbb{R}^E$ definido como:

$$P = \operatorname{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\}$$

Exercício. Seja P o politopo definido acima. Mostre que o conjunto dos vértices de P é precisamente o conjunto dos vetores de incidência dos elementos de $\mathcal{S}$.

Ou seja, o problema original é equivalente a

$$\max\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{x} \in P\}$$

já que as soluções ótimas estão nos vértices de P

Dificuldade

$$P = \mathrm{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\} \qquad \max\{c^T x : x \in P\}$$

Dificuldade

$$P = \mathrm{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\} \qquad \max\{c^T x : x \in P\}$$

Como P é um poliedro, ele tem uma representação externa

Dificuldade

$$P = \mathrm{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\} \qquad \max\{c^T x : x \in P\}$$

Como P é um poliedro, ele tem uma representação externa

Porém, a representação externa pode ser muito grande

Dificuldade

$$P = \text{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\} \qquad \max\{c^T x : x \in P\}$$

Como P é um poliedro, ele tem uma representação externa

Porém, a representação externa pode ser muito grande

- E separar pode ser difícil

Dificuldade

$$P = \mathrm{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\} \qquad \max\{c^T x : x \in P\}$$

Como P é um poliedro, ele tem uma representação externa

Porém, a representação externa pode ser muito grande

- E separar pode ser difícil

Mas nem sempre!

Dificuldade

$$P = \mathrm{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\} \qquad \max\{c^T x : x \in P\}$$

Como P é um poliedro, ele tem uma representação externa

Porém, a representação externa pode ser muito grande

- E separar pode ser difícil

Mas nem sempre!

- Veremos um exemplo em que a representação externa é polinomial

Dificuldade

$$P = \mathrm{conv}\{\boldsymbol{\chi}^S : S \in \mathcal{S}\} \qquad \max\{c^T x : x \in P\}$$

Como P é um poliedro, ele tem uma representação externa

Porém, a representação externa pode ser muito grande

- E separar pode ser difícil

Mas nem sempre!

- Veremos um exemplo em que a representação externa é polinomial
- Outro em que é exponencial, mas pode ser separada em tempo polinomial

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na verdade, um PLI com variáveis binárias é também um problema de otimização combinatória linear

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na verdade, um PLI com variáveis binárias é também um problema de otimização combinatória linear

Se queremos resolver $\max\{\boldsymbol{c}^T\boldsymbol{x} \colon \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^n\}$, basta considerar

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na verdade, um PLI com variáveis binárias é também um problema de otimização combinatória linear

Se queremos resolver $\max\{\boldsymbol{c}^T\boldsymbol{x} \colon \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^n\}$, basta considerar

- $E = \{1, 2, \ldots, n\}$

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na verdade, um PLI com variáveis binárias é também um problema de otimização combinatória linear

Se queremos resolver $\max\{\boldsymbol{c}^T\boldsymbol{x} \colon \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^n\}$, basta considerar

- $E = \{1, 2, \ldots, n\}$
- $c(e) = c_e$

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na verdade, um PLI com variáveis binárias é também um problema de otimização combinatória linear

Se queremos resolver $\max\{\boldsymbol{c}^T\boldsymbol{x}\colon \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^n\}$, basta considerar

- $E = \{1, 2, \ldots, n\}$
- $c(e) = c_e$
- $\mathcal{S} = \{S \subseteq E\colon \boldsymbol{A}\boldsymbol{\chi}^S \leq \boldsymbol{b}\}$

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na verdade, um PLI com variáveis binárias é também um problema de otimização combinatória linear

Se queremos resolver $\max\{\boldsymbol{c}^T\boldsymbol{x} \colon \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^n\}$, basta considerar

- $E = \{1, 2, \ldots, n\}$
- $c(e) = c_e$
- $\mathcal{S} = \{S \subseteq E \colon \boldsymbol{A}\boldsymbol{\chi}^S \leq \boldsymbol{b}\}$

Ou seja, problemas de otimização combinatória linear é uma classe equivalente a PLI com variáveis binárias

Na outra direção

Vimos que problemas de otimização combinatória linear podem ser formulados como problemas de programação linear (inteira)

Na verdade, um PLI com variáveis binárias é também um problema de otimização combinatória linear

Se queremos resolver $\max\{\boldsymbol{c}^T\boldsymbol{x} \colon \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^n\}$, basta considerar

- $E = \{1, 2, \ldots, n\}$
- $c(e) = c_e$
- $\mathcal{S} = \{S \subseteq E \colon \boldsymbol{A}\boldsymbol{\chi}^S \leq \boldsymbol{b}\}$

Ou seja, problemas de otimização combinatória linear é uma classe equivalente a PLI com variáveis binárias

Mas o que aprendemos/aprenderemos é útil também para PLI com variáveis não binárias