Programação Inteira, 2023.1

Relaxação Lagrangiana

DEMA/UFC

Overleaf

Entrada: problema (P)
Saída: solução inteira x^*

1. Iniciar estrutura de armazenamento de subproblemas: $Q = \{\emptyset\}$
2. Enquanto $Q \neq emptyset$:
    - Tome $s \in Q$
    - $(z^R, x^R) :=$ relaxação(P) satisfazendo s
    - Se relaxação for inviável, `continue`
    - Se $x^* \neq \emptyset$ e $z^R \leq z(x^*)$, senão `continue`
    - Se x^R é solução inteira, considere atualizar x^*, senão:
        - Considere produzir solução inteira viável e atualizar x^*
        - Escolha x_j^R variável fracionária
        - $Q := Q \cup \{s \cup \{x_j \leq \lfloor x_j^R \rfloor\}, s \cup \{x_j \geq \lceil x_j^R \rceil\}\}$
3. Retorne x^*.

Relaxação Lagrangiana

$$
\begin{aligned}
(P) \quad z = \max \quad & c^\top x && (1)\\
\text{s.a:} \quad & \\
& Dx \leq d && (2)\\
& x \in X, && (3)
\end{aligned}
$$

$$(LR(u)) \quad z(u) = \max \quad c^\top x + u^\top (d - Dx) \tag{4}$$

s.a:

$$x \in X. \tag{5}$$

$$(LD) \quad w_{LD} = \min\{z(u) : u \geq 0\}. \tag{6}$$

$$w_{LD} = \min_{u \geq 0} \left\{ \max\{c^\top x^i + u^\top (d - Dx^i) : i = 1, \ldots, t\} \right\}. \tag{7}$$

$$\begin{aligned} w_{LD} = \min \quad & q & (8)\\ \text{s.a:} \quad & & \\ & q \geq c^\top x^i + u^\top (d - Dx^i), \quad i = 1, \ldots, t & (9)\\ & q \in \mathbb{R}, \quad u \in \mathbb{R}^m_+. & (10) \end{aligned}$$

Um vetor $y \in \mathbb{R}^m$ é um subgradiente de uma função convexa f no ponto $\bar{x}$ se

$$f(x) \geq f(\bar{x}) + y^\top (x - \bar{x}).$$

Vamos aplicar esta ideia à função $z(u)$.

Dado $\hat{u}$, seja $\hat{x}$ a solução ótima de $(LR_{\hat{u}})$ com valor $z(\hat{u})$. Podemos afirmar que $y = d - D\hat{x}$ é um subgradiente de $z(u)$ no ponto $\hat{u}$:

$$\begin{aligned}
z(\hat{u}) + y^\top (u - \hat{u}) &= z(\hat{u}) + (d - D\hat{x})^\top (u - \hat{u}) \\
&= c^\top \hat{x} + \hat{u}^\top (d - D\hat{x}) + (d - D\hat{x})^\top u - (d - D\hat{x})^\top \hat{u} \\
&= c^\top \hat{x} + (d - D\hat{x})^\top u \\
&\leq \max\{c^\top x + (d - Dx)^\top u : x \in X\} \\
&= z(u).
\end{aligned}$$

Algoritmo de subgradientes

1. Escolha u^0 como multiplicadores iniciais
2. Para $k = 0, \ldots, K - 1$:
   - Resolver (LR_{u^k}); seja x^k a solução ótima e $z(u^k)$ seu valor
   - Calcular subgradiente $(d - Dx^k)$
   - Calcular passo $t^k = \dfrac{\lambda^k \left(z(u^k) + \underline{Z}\right)}{\|Dx^k - d\|^2}$
   - Definir $u^{k+1} = \left[u^k + t^k \left(Dx^k - d\right)\right]_+$

- $0 < \lambda^k \leq 2$ é um parâmetro que tipicamente inicia como $\lambda^0 = 2$ e é reduzido pela metade sempre que o valor de $z(u^k)$ não é reduzido ao longo de certo número de iterações;
- A notação $[a]_+ = max(a, 0)$ denota a "parte positiva" de a. No algoritmo, estamos nos certificando de que componentes negativas do vetor são substituídas por zero;
- $\underline{Z}$ é um limite inferior para o valor ótimo de (P), obtido, por exemplo, via uma heurística.

Algoritmo de subgradientes

1. Como o dual lagrangeano é de minimização, calculamos u^{k+1} usando o subgradiente multiplicado por -1, isto é, $(Dx^k - d)$ em vez de $(d - Dx^k)$.
2. A não ser que obtenhamos u^k tal que $z(u^k)$ seja igual ao valor de uma solução viável de (P), não podemos provar a otimalidade de uma solução obtida pelo algoritmo de subgradientes.

3. Há certas garantias que asseguram a convergência do algoritmo (por exemplo, $\displaystyle\lim_{k\to\infty} t^k = 0$ e $\displaystyle\sum_{i=1}^{k} t^i \to \infty$) mas o algoritmo anterior que mostramos costuma apresentar melhor desempenho prático.
4. Se usarmos uma heurística para produzir soluções viáveis a partir da solução do problema relaxado, temos o que é chamado de uma *heurística lagrangiana*.