Algoritmos de Aproximación

Clase 6
Esquema Primal-Dual y Ajuste Dual

Pablo Romero

Lunes 28 de agosto de 2023. Montevideo, Uruguay.

Problema Primal

Sea Π un COP que se puede formular mediante programación lineal entera de modo que su relajación es el siguiente *Problema Primal*:

$$\min_{x} \sum_{j=1}^{n} c_j x_j \tag{1}$$

$$s.a. \ \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \ i = 1, \ldots, m. \tag{2}$$

$$x_j \geq 0, \ j = 1, \ldots, n, \tag{3}$$

donde $x = (x_1, x_2, \ldots, x_n)$ es *primal factible* si satisface (2) y (3). Dada una instancia I de Π denotaremos $OPT_f(I)$ al mínimo global. En general, $OPT_f(I) \leq OPT(I)$.

Problema Dual

Dado un problema primal que satisface la forma anterior, su correspondiente problema dual se define de la siguiente manera:

$$\max_{y} \sum_{i=1}^{m} b_i y_i \tag{4}$$

$$s.a. \sum_{i=1}^{m} a_{ij} y_i \leq c_j,\ j = 1, \ldots, n. \tag{5}$$

$$y_i \geq 0,\ i = 1, \ldots, m, \tag{6}$$

donde $y = (y_1, \ldots, y_m)$ es *dual factible* si satisface (5) y (6).

Dualidad Débil

Teorema 1 (Dualidad débil).

Si $x = (x_1, x_2, \ldots, x_n)$ *es una solución primal factible e* $y = (y_1, y_2, \ldots, y_n)$ *es una solución dual factible, entonces* $\sum_{i=1}^{m} b_i y_i \leq \sum_{j=1}^{n} c_j x_j$.

Prueba. Con la notación del enunciado tenemos que:

$$\sum_{i=1}^{m} b_i y_i \leq \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} y_i x_j \leq \sum_{j=1}^{n} c_j x_j.$$

donde hemos utilizado las restricciones (2) en la primera desigualdad y las restricciones (5) en la segunda desigualdad. ■

Esquema Primal-Dual

Teorema 2.

Sea Π *un COP formulado mediante programación lineal entera y sean* $x^* = (x_1^*, \ldots, x_n^*)$ *e* $y^* = (y_1^*, \ldots, y_m^*)$ *soluciones primal y dual factibles tales que:*

1. *Para cada* $j \in \{1, 2, \ldots, n\}$, $x_j^* = 0$ *o* $c_j \leq \alpha \sum_{i=1}^{m} a_{ij} y_i^* \leq \alpha c_j$.
2. *Para cada* $i \in \{1, 2, \ldots, m\}$, $y_i^* = 0$ *o* $b_i \leq \sum_{j=1}^{n} a_{ij} x_j^* \leq \beta b_i$.

Entonces, $\sum_{i=1}^{n} c_j x_j^* \leq \alpha\beta OPT$.

Prueba.

$$\sum_{j=1}^{n} c_j x_j^* \leq \alpha \sum_{j=1}^{n} \sum_{i=1}^{m} (a_{ij} y_i^*) x_j^* \leq \alpha\beta \sum_{i=1}^{m} b_i y_i^* \leq \alpha\beta OPT,$$

donde hemos utilizado (1) y (2) en las dos primeras desigualdades y el Teorema 1 en la penúltima desigualdad. ■

Cubrimiento de Conjuntos - Esquema Primal-Dual

Es posible formular cada instancia $I = (\mathcal{S}, U, c)$ del cubrimiento de conjuntos mediante el problema de programación lineal entera:

$$\begin{aligned} \min_{x_S} \sum_{S \in \mathcal{S}} c(S) x_S \\ s.a. \sum_{S : e \in S} x_S \geq 1, \forall e \in U. \\ x_S \in \{0, 1\}, \forall S \in \mathcal{S}, \end{aligned}$$

donde $x_S = 1$ si $S \in \mathcal{U}$ o 0 en caso contrario. Para cada elemento u de U definimos $f_u = |\{S \in \mathcal{S} : u \in S\}|$ y $f = \max_{u \in U}\{f_u\}$. Construiremos un algoritmo de aproximación para el cubrimiento de conjuntos de factor f utilizando el Teorema 2.

Cubrimiento de Conjuntos - Problema Primal

La relajación del problema de programación lineal entera es el siguiente problema primal:

$$\begin{aligned} \min_{x_S} \sum_{S \in \mathcal{S}} & c(S) x_S \\ s.a. \sum_{S : e \in S} x_S & \geq 1, \forall e \in U. \\ x_S & \geq 0, \forall S \in \mathcal{S}. \end{aligned}$$

Cubrimiento de Conjuntos - Problema Dual

Su correspondiente problema dual es el siguiente:

$$\max_{y} \sum_{e \in U} y_e \tag{7}$$

$$s.a. \sum_{e: e \in S} y_e \leq c(S), \ \forall S \in \mathcal{S} \tag{8}$$

$$y_e \geq 0, \ \forall e \in U. \tag{9}$$

Si queremos utilizar el esquema Primal-Dual con $\alpha = 1$ y $\beta = f$ debemos lograr las siguientes condiciones:

- $\forall S \in \mathcal{S}, (x_S \neq 0 \rightarrow \sum_{e: e \in S} y_e = c(S));$
- $\forall e \in U, (y_e \neq 0 \rightarrow \sum_{S: e \in S} x_S \leq f).$

Por la definición de f surge que las segundas restricciones siempre son satisfechas.

Algoritmo Primal-Dual

Algoritmo 1 $\mathcal{U} = PrimalDual(\mathcal{S}, U, c)$

$\mathcal{U} \leftarrow \emptyset$
$C \leftarrow \emptyset$
$x \leftarrow 0$
$y \leftarrow 0$
while $C \neq U$ **do**
$y' \leftarrow Aumentar(y, e)$ {hasta que $\exists S' \in \mathcal{S}, \sum_{e:e\in S} y_e = c(S)$}
$\mathcal{V} \leftarrow Saturados(y')$ {retorna $\{S' \in \mathcal{S}, \sum_{e:e\in S'} y_e = c(S')\}$}
$C \leftarrow C \cup \bigcup_{S_i \in \mathcal{V}} S_i$
$\mathcal{U} \leftarrow \mathcal{U} \cup \mathcal{V}$
for all $S \in \mathcal{U}$ **do**
$x_S \leftarrow 1$
end for
end while
return $\mathcal{U}$

Algoritmo Primal-Dual

Teorema 3.

PrimalDual(I) *es de factor* f *para el cubrimiento de conjuntos.*

Prueba. Sea I una instancia del cubrimiento de conjuntos. Al inicio y es dual factible, y en cada ciclo **while** se satisfacen las restricciones (8) y (9). Tal ciclo **while** se ejecuta mientras no hayan elementos de U cubiertos. Además en cada ciclo se cubre algún elemento no cubierto, por lo que el ciclo termina en no más de $|U|$ iteraciones. En su salida, el conjunto $\mathcal{U}$ es cubrimiento, y por lo tanto x es primal factible y entero. En cada iteración se define $x_S = 1$ por cada conjunto S saturado. Entonces se respetan las hipótesis del Teorema 2, y *PrimalDual*(I) retorna un cubrimiento $\mathcal{U}$ tal que $\sum_{S \in \mathcal{U}} c(S) = \sum_{S \in \mathcal{S}} c_S x_S \leq f OPT_\Pi(I)$. ■

Ajuste Dual en el Cubrimiento de Conjuntos

Otra técnica para construir algoritmos de aproximación se basa en tener una solución factible del dual y se denomina *ajuste dual*. Recudimos precios en el dual como sigue.

Lema 1.

La tupla y tal que $y_e = \frac{p(e)}{H_n}$ para cada $e \in U$ es dual factible.

Prueba. Sea $S_j \in \mathcal{S}$ cualquiera cuyos elementos $e_1, e_2, \ldots, e_k$ se ordenan según Greedy. Al cubrir e_i quedan al menos $k - i + 1$ elementos. Como S_j cubre e_i a un costo no mayor que $c(S_j)/(k - i + 1)$ entonces $p(e_i) \leq c(S)/(k - i + 1)$, y

$$\sum_{s \in S_j} y_e \leq \frac{c(S_j)}{H_n} \sum_{i=1}^{k} \frac{1}{k - i + 1} = \frac{H_k}{H_n} c(S_j) \leq c(S_j),$$

por lo que se cumplen (8) y (9). ■

Ajuste Dual en el Cubrimiento de Conjuntos

Teorema 4.

Greedy es un algoritmo de aproximación de factor H_n para el cubrimiento de conjuntos.

Prueba. Sea $I = (\mathcal{S}, U, c)$ una instancia cualquiera del cubrimiento de conjuntos. Ya vimos que Greedy es un algoritmo. Sea $\mathcal{U} = Greedy(I)$. También vimos que el costo de $\mathcal{U}$ es la suma de los precios de todos sus elementos. Entonces:

$$c(\mathcal{U}) = \sum_{e \in U} p(e) = H_n \sum_{e \in U} y_e \leq H_n OPT_f(I) \leq H_n OPT(I),$$

donde en la primera desigualdad usamos el Lema 1. ■

Gap de integralidad

Definción 1.

Sea Π un COP que se formula mediante un ILP de modo que su valor óptimo es $OPT_{\Pi}(I)$. Para cada instancia I de Π, denotemos $OPT_f(I)$ al valor óptimo del problema relajado. La discrepancia o gap de integralidad es el siguiente cociente:

$$g = sup_{I \in \mathcal{I}} \frac{OPT_{\Pi}(I)}{OPT_f(I)}$$

Observaciones

- En COPs $\mathcal{NP}$-difíciles se tiene que $g > 1$.
- Con familias justas se muestra que los análisis anteriores no se pueden mejorar.

Comentarios

1. La mayoría de los algoritmos de aproximación se inspiran en programación lineal.
2. Mediante la dualidad fuerte se pueden recuperar las identidades minimax.
3. Veremos la aplicación del esquema primal-dual al problema de Steiner y generalizaciones.
4. Antes estudiaremos algoritmos de aproximación que se basan en redondeo y aleatorización para el cubrimiento de conjuntos.