Algoritmos de Aproximación

Clase 3
Concepto de Algoritmo de aproximación

Pablo Romero

Lunes 14 de agosto de 2023. Montevideo, Uruguay.

Problemas de Optimización Combinatoria

Problemas de Optimización Combinatoria (COP)

- Un COP es una terna $\Pi = (D_\Pi, S_\Pi, f)$, donde D_Π es una colección de *instancias del problema* Π, para cada instancia I tenemos un conjunto finito $S_\Pi(I)$ de *soluciones factibles*, y f es la *función objetivo*, que para cada instancia I y cada solución factible s de I asigna un número racional $f(I, s)$.
- Dado un problema Π y una instancia $I \in D_\Pi$, decimos que $x \in S_\Pi(I)$ es *óptimo global de la instancia* I si para todo $y \in S_\Pi(I)$ se cumple que $f(I, x) \leq f(I, y)$.
- Si $I \in D_\Pi$, denotamos $OPT_\Pi(I) = f(I, x) = \min_{s \in S_\Pi(I)} f(I, s)$.

Pregunta

¿Dado un COP, es posible hallar para toda instancia I el valor óptimo $OPT_\Pi(I)$?

Ejemplos de COP (1)

Cubrimiento de vértices de mínimo cardinal (CVC)

Dado un grafo $G = (V, E)$, el CVC consiste en hallar el subconjunto $V' \subseteq V$ que cumple que $e \in V' \neq \emptyset$ tal que $|V'|$ sea mínimo.

Entonces, el CVC es el problema $\Pi_1 = (\mathcal{G}, S_{\Pi_1}, f_1)$, donde

- $\mathcal{G}$ es la colección de todos los grafos simples.
- $S_{\Pi_1}(G) = \{V' : V' \subseteq V, \forall e \in E, V' \cap \{e\} \neq \emptyset\}$.
- $f_1(V') = |V'|$.
- $OPT_{\Pi_1}(G) = \min_{\{V' \in S_{\Pi_1}(G)\}} |V'|$.

Ejemplos de COP (2)

Cubrimiento de conjuntos (SC)

Dado un universo U con n elementos, una colección $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$ de subconjuntos que cubren U y una función de costos $c : \mathcal{S} \to \mathbb{Q}^+$, seleccionar una subcolección $\mathcal{U} \subseteq \mathcal{S}$ que cubra U con costo mínimo.
Sea $\mathcal{X}$ el conjunto que contiene a todas las colecciones de cubrimientos $\mathcal{S}$ de U. Entonces, el SC es el problema $\Pi_2 = (\mathcal{I}_2, S_{\Pi_2}, f_2)$, donde

- $\mathcal{I}_2 = \{(\mathcal{S}, U, c) : \mathcal{S} \in \mathcal{X}, c : \mathcal{S} \to \mathbb{Q}^+\}$.
- $S_{\Pi_2}(\mathcal{S}, U) = \{\mathcal{U} : \mathcal{U} \subseteq \mathcal{S}, U \in \mathcal{X}\}$.
- $f_2((\mathcal{S}, U, c), \mathcal{U}) = \sum_{S_i \in \mathcal{U}} c(S_i)$.
- $OPT_{\Pi_2}(\mathcal{S}, U, c) = \min_{\{\mathcal{U} \in S_{\Pi_2}(\mathcal{S}, U, c)\}} \sum_{S_i \in \mathcal{S}} c(S_i)$.

Ejemplos de COP (3)

Problema de Steiner (STP)

Dado un grafo $G = (V, E)$, una partición $V = R \cup S$ y costos en las aristas $c : E \to \mathbb{Q}^+$, el STP consiste en hallar el árbol de Steiner T que cubre a todos los vértices requeridos de R (y que contenga cualquier subconjunto de vértices de Stieiner S)
Sea $\mathcal{T}$ la clase de todos los árboles. Entonces, el STP es el problema $\Pi_3 = (\mathcal{I}_3, S_{\Pi_3}, f_3)$, donde

- $\mathcal{I}_3 = \{(G, R, c) : G \in \mathcal{G}, G = (V, E), R \subseteq V, c : E \to \mathbb{Q}^+\}$.
- $S_{\Pi_3}(G, R, c) = \{T \in \mathcal{T} : T \subseteq G, R \subseteq V(T)\}$.
- $f_3((G, R, c), T) = \sum_{e \in E(T)} c(e)$.
- $OPT_{\Pi_3}((G, R, c)) = \min_{\{T \in S_{\Pi_3}(G,R,c)\}} \sum_{e \in E(T)} c(e)$.

Ejemplos de COP (4)

Problema del vendedor ambulante (TSP)

Dado un grafo completo con al menos 3 vértices y con costos racionales no negativos en sus aristas, el TSP consiste en hallar un ciclo hamiltoniano en tal grafo con costo mínimo.
Entonces, el TSP es el problema $\Pi_4 = (\mathcal{I}_4, S_{\Pi_4}, f_4)$, donde

- $\mathcal{I}_4 = \{(K_n, c) : n \in \mathbb{Z}^+, n \geq 3, c : E(K_n) \to \mathbb{Q}^+ \cup \{0\}\}$.
- $S_{\Pi_4}((K_n, c)) = \{\mathcal{C} \subseteq K_n : \mathcal{C} \cong C_n\}, \forall I \in \mathcal{I}_=4$.
- $f_4((K_n, c), \mathcal{C}) = \sum_{e \in E(\mathcal{C})} c(e)$.
- $OPT_{\Pi_4}((K_n, c)) = \min_{\{\mathcal{C} : \mathcal{C} \subseteq K_n\}} \sum_{e \in E(\mathcal{C})} c(e)$.

Ejemplos de COP (5)

Red 2-conexa de mínimo costo (MWTCSN)

Dado un grafo completo K_n con costos métricos en sus aristas $c : E(K_n) \to \mathbb{Q}^+ \cup \{0\}$, el MWTCSN consiste en hallar el subgrafo 2-conexo de K_n con mínimo costo.

Sea $\mathcal{G}^{(2)}$ la clase de grafos 2-conexos. Entonces, el MWTCSN es el problema $\Pi_5 = (\mathcal{I}_5, S_{\Pi_5}, f_5)$, donde

- $\mathcal{I}_5 = \{(K_n, c) : n \in \mathbb{Z}^+, c : E(K_n) \to \mathbb{Q}^+, c(xz) \leq c(xy) + c(yz)\}$.
- $S_{\Pi_5}((K_n, c)) = \{G \subseteq K_n : G \in \mathcal{G}^{(2)}$.
- $f_5((K_n, c), G) = \sum_{e \in E(G)} c(e)$.
- $OPT_{\Pi_4}(K_n, c) = \min_{\{\mathcal{C} \subseteq K_n\}} \sum_{e \in E(\mathcal{C})} c(e)$.

Versión de decisión de un COP

Defición 1.

Un problema de decisión es un par $X = (D_X, p)$*, donde* D_X *es un conjunto no vacío de instancias de decisión y* p *es una proposición definible sobre* D_X*, es decir que* $p : D_X \to \{0, 1\}$*. El conjunto de instancias positivas de* X *es* $D_X^+ = \{x \in D_X : p(x) = 1\}$*.*

Siempre es posible convertir un COP a un problema de decisión.

Defición 2.

Para cada COP $\Pi = (D_\Pi, S_\Pi, f)$ *y cada* $k \in \mathbb{Q}^+$*, definimos* $X_\Pi^k = (D_\pi, p_k)$*, donde* $p_k : D_\Pi \to \{0, 1\}$ *es tal que* $p_k(I) = \{\exists s \in S_\Pi(I) : f(s, I) \leq k\}$*.*

Nociones de Complejidad

Asumiremos el modelo de computación clásico. Sea $X = (D_X, p)$ un problema de decisión y $\mathcal{C} : D_X \to \{0,1\}^*$ alguna codificación.

Nociones de Complejidad

- Dada $I \in D_X$, la cantidad de bits de $\mathcal{C}(I)$ se denota $|I|$ y es el *tamaño* de I.
- La codificación de todas las instancias positivas de X es el *lenguaje de* X, es decir, $\mathcal{L}(X) = \mathcal{C}(D_X^+)$.
- El problema X pertenece a la clase $\mathcal{P}$ si existe algún algoritmo $\mathcal{A}$ de tiempo polinomial tal que $\mathcal{A}(I) = p(I)$ para toda $I \in D_X$.
- Un problema de decisión X pertenece a la clase $\mathcal{NP}$ si existe un algoritmo de tiempo polinomial $\mathcal{A}$ que permite certificar que $\mathcal{A}(I) = 1$ para toda $I \in D_X^+$. En tal caso, el algoritmo $\mathcal{A}$ se llama *certificado positivo*.

Completitud y Reducibilidad de Karp

Sean X e Y dos problemas de decisión.

Completitud y Reducibilidad de Karp

- Una reducción R de un problema X a Y es una función R computable en tiempo polinomial tal que $x \in \mathcal{L}(X)$ si y sólo si $R(x) \in \mathcal{L}(Y)$. Denotamos $X \subseteq_K Y$.
- Un problema Y es $\mathcal{C}$-difícil si $\forall X \in \mathcal{C}, X \subseteq_K Y$.
- Un problema Y es $\mathcal{C}$-completo si es $\mathcal{C}$-difícil y además $Y \in \mathcal{C}$.

Método de demostración de $\mathcal{NP}$-Completitud

Sea Y un problema $\mathcal{NP}$-difícil cualquiera. Luego, X es $\mathcal{NP}$-completo si tiene algún certificado positivo $\mathcal{A}$ y además $X \subseteq_K Y$.

Problema de optimización Combinatoria $\mathcal{NP}$-difícil

Un problema de $\mathcal{NP}$-Optimización es un COP $\Pi = (D_\Pi, S_\Pi, f)$ que respeta todas las condiciones siguientes:

- Existe una codificación polinomial $\mathcal{C} : D_\Pi \to \{0,1\}^*$.
- Para cada instancia $I \in D_\Pi$, el conjunto $S_\Pi(I)$ es no vacío.
- Dado un par (s, I), es posible decidir en tiempo polinomial si $s \in D_\pi(I)$.
- Para cada instancia I y cada solución factible $s \in S_\Pi(I)$, existe algún polinomio q tal que $|s| \leq q(|I|)$.
- Para cada instancia I y cada solución factible $s \in S_\Pi(I)$, la función $f(s, I)$ se puede evaluar en tiempo polinomial.

Definción 3.

Un COP es $\mathcal{NP}$-difícil si es un problema de $\mathcal{NP}$-optimización y además su correspondiente versión de decisión es $\mathcal{NP}$-difícil.

Algoritmo de aproximación

Defínción 4.

*Dado un COP $\mathcal{NP}$-difícil $\Pi = (D_\Pi, S_\Pi, f)$ y una función $\delta : \mathbb{Z}^+ \to \mathbb{Q}^+$, decimos que $\mathcal{A}$ es un **algoritmo de aproximación de factor δ** si para cada $I \in D_\Pi$, la salida del algoritmo $s = \mathcal{A}(I)$ verifica que $s \in S_\Pi(I)$ y además que $f(I, s) \leq \delta(I) OPT_\Pi(I)$.*

Defínción 5.

La clase APX consiste de todos los problemas de optimización combinatoria Π que admiten algún algoritmo de aproximación de factor α constante.

Reducción que preserva el factor de aproximación

Definción 6.

Sean $\Pi_1 = (\mathcal{I}_1, S_{\Pi_1}, c_1)$ y $\Pi_2 = (\mathcal{I}_2, S_{\Pi_2}, c_2)$ dos COPs.
*Una **reducción que preserva el factor de aproximación** consiste de un par de algoritmos de tiempo polinomial (g_1, g_2) tales que:*

- $g_1 : \mathcal{I}_1 \to \mathcal{I}_2$ *cumple que si* $I_2 = f(I_1)$ *entonces* $OPT_{\Pi_2}(I_2) \leq OPT_{\Pi_1}(I_1)$.
- *Para toda* $I_2 \in \mathcal{I}_2$ *se cumple que* $s = g_2(I_2, t) \in S_{\Pi_1}(I_1)$ *y además* $c_1(I_1, s) \leq c_2(I_2, t)$.

Ejercicio 2

Probar que si (g_1, g_2) es una reducción de Π_1 a Π_2 que preserva el factor de aproximación y $\mathcal{A}$ es un algoritmo de aproximación de factor α para Π_2, entonces $\mathcal{A}' = g_2 \circ \mathcal{A} \circ g_1$ es un algoritmo de aproximación de factor α para Π_1.