Chapter 7: The Primal-Dual Method

(cp. Williamson & Shmoys, Chapter 7)

Set Cover Problem

Given: A set of elements $E = \{e_1, \ldots, e_n\}$, a family of subsets $\{S_1, \ldots, S_m\} \subseteq 2^E$, and a weight $w_j \geq 0$ for each $j \in \{1, \ldots, m\}$.

Task: Find $I \subseteq \{1, \ldots, m\}$ minimizing $\sum_{j \in I} w_j$ such that $\bigcup_{j \in I} S_j = E$.

LP relaxation:

$$\begin{array}{lrcll} \min & \sum_{j=1}^{m} w_j \cdot x_j & & & \\ \text{s.t.} & \sum_{j : e_i \in S_j} x_j & \geq & 1 & \text{for all } i = 1, \ldots, n \\ & x_j & \geq & 0 & \text{for all } j = 1, \ldots, m \end{array}$$

Dual LP:

$$\begin{array}{lrcll} \max & \sum_{i=1}^{n} y_i & & & \\ \text{s.t.} & \sum_{e_i \in S_j} y_i & \leq & w_j & \text{for all } j = 1, \ldots, m \\ & y_i & \geq & 0 & \text{for all } i = 1, \ldots, n \end{array}$$

Primal-Dual Algorithm for Set Cover Problem (see Ch. 1)

1. set $y :\equiv 0$ and $I := \emptyset$;
2. while $\exists e_k \notin \bigcup_{j\in I} S_j$
3. increase y_k until $\exists j$ with $e_k \in S_j$ such that $\sum_{i:e_i\in S_j} y_i = w_j$;
4. set $I := I \cup \{j\}$;

Theorem ??

The primal-dual algorithm is an f-approximation algorithm for the Set Cover Problem where $f := \max_{i=1,\dots,n} |\{j \mid e_i \in S_j\}|$.

Proof:

$$\sum_{j\in I} w_j = \sum_{j\in I}\sum_{i:e_i\in S_j} y_i = \sum_{i=1}^{n} y_i \cdot |\{j \in I \mid e_i \in S_j\}|$$

$$\le f \cdot \sum_{i=1}^{n} y_i \le f \cdot \mathrm{OPT}_{LP} \le f \cdot \mathrm{OPT}$$

□

Approximate Complementary Slackness

Remark. The pair of feasible solutions (x, y) to the primal and dual LP found by the algorithm satisfies

$$x_j > 0 \implies \sum_{e_i\in S_j} y_i = w_j \qquad (compl.\ slackness)$$

$$y_i > 0 \implies \sum_{j:e_i\in S_j} x_j \le f \qquad (approx.\ compl.\ slackness)$$

The analysis on the previous slide only relies on these two properties!

Feedback Vertex Set Problem

Given: Undirected graph $G = (V, E)$ with node weights $w_i \geq 0$, $i \in V$.

Task: Find $S \subseteq V$ minimizing $\sum_{i \in S} w_i$ such that $G[V \setminus S]$ is acyclic.

Integer programming formulation: Let $\mathcal{C}$ denote the set of all cycles in G.

$$\begin{aligned} \min \quad & \sum_{i \in V} w_i \cdot x_i & \\ \text{s.t.} \quad & \sum_{i \in C} x_i \geq 1 & \text{for all } C \in \mathcal{C}, \\ & x_i \in \{0, 1\} & \text{for all } i \in V. \end{aligned}$$

Dual of LP relaxation ($x \geq 0$):

$$\begin{aligned} \max \quad & \sum_{C \in \mathcal{C}} y_C & \\ \text{s.t.} \quad & \sum_{C \in \mathcal{C} :\, i \in C} y_C \leq w_i & \text{for all } i \in V, \\ & y_C \geq 0 & \text{for all } C \in \mathcal{C}. \end{aligned}$$

Primal-Dual Algorithm for Feedback Vertex Set Problem

1. set $y :\equiv 0$ and $S := \emptyset$;
2. while there is a cycle C in G
3. increase y_C until there is an $i \in C$ with $\sum_{C' : i \in C'} y_{C'} = w_i$;
4. set $S := S \cup \{i\}$ and delete i from G;
5. repeatedly remove nodes of degree one from G;

Analysis:

$$\sum_{i \in S} w_i = \sum_{i \in S} \sum_{C :\, i \in C} y_C = \sum_{C \in \mathcal{C}} |S \cap C| \cdot y_C$$

- Idea: If $|S \cap C| \leq \alpha$ whenever $y_C > 0$, we get performance ratio α.
- But: If we choose arbitrary C in each iteration, $|S \cap C|$ can be large.
- Idea: Always choose short cycle C with $|C| \leq \alpha$.
- But: This is not always possible (e. g., if graph is one large cycle).
- Idea: From path of nodes of degree two, algorithm chooses ≤ 1 node.

Refined Primal-Dual Algorithm for Feedback Vertex Set

Lemma 7.1.

In any graph G that has no nodes of degree one, there is a cycle with $\leq 2\lceil \log_2 n \rceil$ nodes of degree 3 or more, and it can be found in linear time.

Proof:. . . □

Theorem 7.2.

If the primal-dual algorithm chooses in each iteration a cycle with at most $\leq 2\lceil \log_2 n \rceil$ nodes of degree 3 or more, it has performance ratio $4\lceil \log_2 n \rceil$.

Proof:. . . □

Remarks.

- The LP relaxation has an integrality gap of $\Omega(\log n)$.
- There is a primal-dual 2-approximation algorithm based on a more sophisticated integer programming formulation

Shortest s-t-Path Problem

Given: Undir. graph $G = (V, E)$ with edge costs $c_e \geq 0$, $e \in E$; $s, t \in V$

Task: Find minimum-cost s-t-path.

IP formulation: (let $\mathcal{S} := \{S \subseteq V \mid s \in S,\ t \in V \setminus S\}$)

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e \cdot x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(S)} x_e \geq 1 && \text{for all } S \in \mathcal{S}, \\ & x_e \in \{0,1\} && \text{for all } e \in E. \end{aligned}$$

Dual of LP relaxation ($x \geq 0$):

$$\begin{aligned} \max \quad & \sum_{S \in \mathcal{S}} y_S \\ \text{s.t.} \quad & \sum_{S \in \mathcal{S}:\, e \in \delta(S)} y_S \leq c_e && \text{for all } e \in E, \\ & y_S \geq 0 && \text{for all } S \in \mathcal{S}. \end{aligned}$$

Primal-Dual Algorithm for Shortest s-t-Path Problem

1. set $y :\equiv 0$ and $F := \emptyset$;
2. while there is no s-t-path in F
3. let C be the connected component of (V, F) containing s;
4. increase y_C until there is an $e \in \delta(C)$ with $\sum_{S \in \mathcal{S}:\, e \in \delta(S)} y_S = c_e$;
5. set $F := F \cup \{e\}$;
6. delete edges from F that do not lie on s-t-path in F;

Lemma 7.3.

Throughout the algorithm, the set of edges in F always forms a tree containing node s. □

Theorem 7.4.

The algorithm finds a shortest s-t-path.

Proof:. . . □