TD8: Primal-Dual Algorithms

Exercise 1. Maximum Matching in Bipartite Graphs
Let $G = (V, E)$ be a bipartite graph.

1. Show that the size of a maximum cardinality matching in G equals the size of the minimum vertex cover.

2. Give an efficient algorithm to find a minimum vertex cover in G. (Hint: Use complementary slackness.)

Exercise 2. The Hungarian Method for the Assignment Problem
Let $G = (V, E)$ be a bipartite graph whose edge costs are nonnegative integers. There is a bipartition $V = (A, B)$ where $|A| = |B| = n$ and the goal is to assign each element in A (e.g. people) to a unique element in B (e.g. tasks) so as to minimize the total cost of the assignment. In other words, we want to find a minimum cost perfect matching between A and B. The goal of this exercise is to study the following primal-dual algorithm for this problem. Let C denote the $n \times n$ cost matrix, where rows are indexed by vertices in A and columns are indexed by vertices in B.

1. For each row in C, decrease each value by the cost of the minimum entry in the row. (Then do the same for each column.) Call the resulting cost matrix $\bar{C}$. Let G_0 denote the subgraph of G that consists of edges in G whose cost in $\bar{C}$ is zero, i.e. $\bar{c}_{ij} = 0$.

2. Find a maximum cardinality matching in G_0. If this matching has size n, terminate the algorithm.

3. Otherwise, find a minimum vertex cover in G_0. Let $A' \subset A, B' \subset B$ denote the vertices in the vertex cover. Note that $|A'| + |B'| < n$.

4. Let $\alpha = \min_{(i,j): i \notin A', j \notin B'} \bar{c}_{ij}$. Subtract α from every row in $\bar{C}$ that is not in A' and add α to each column in B'. Set $C := \bar{C}$ and goto Step 1.

We will now analyze this algorithm.

a) Apply this algorithm to the following 5 by 5 matrix.

$$\begin{pmatrix} 2 & 3 & 4 & 6 & 8 \\ 5 & 5 & 7 & 2 & 3 \\ 6 & 3 & 1 & 2 & 2 \\ 7 & 5 & 4 & 3 & 6 \\ 8 & 7 & 5 & 3 & 2 \end{pmatrix}$$

b) Prove that Step 1 of the algorithm does not affect (i.e. change) the optimal assignment.

c) Show that the maximum matching in Step 2 can be found efficiently.

d) Show that the minimum vertex cover in Step 3 can be found efficiently.

e) Prove that the algorithm terminates.

f) Write the primal and dual linear programs for the assignment problem.

g) Interpret the above algorithm as a primal-dual algorithm.

h) Prove that the final solution is a minimum cost perfect matching by providing a dual certificate.

Exercise 3. Primal-Dual and Dijkstra's Algorithm

Prove that the primal-dual algorithm for shortest s-t-path is equivalent to Dijkstra's algorithm. That is, in each step, it adds the same edge Dijkstra's algorithm would add.

Exercise 4. Shortest s-t-path Tree

Show that the primal-dual algorithm for shortest s-t-path returns a (possible partial) shortest path tree rooted at s before pruning.

Exercise 5. Minimum Cost Arborescence

Given a (strongly connected) directed graph $G = (V, A)$ and a root vertex $r \in V$, an *arborescence* is a subset of edges $S \subseteq A$ such for each vertex $v \in V$, S contains a directed path from r to v. Suppose that each edge $ij \in A$ has a cost $c_{ij} \geq 0$. The *minimum cost arborescence problem* is to find an arborescence in G of minimum cost.

1. Write down the integer program for the minimum cost arborescence problem.
2. Relax the integrality constraint in the integer program to obtain a linear programming relaxation. Write the dual for this linear program.
3. Give a primal-dual algorithm for the minimum cost arborescence problem. (Use the same framework as for the s-t-shortest path. For the pruning stage, delete edges in the reverse order they were added.)
4. Prove that this algorithm is optimal.