CS581 Homework 8 Solution

March 23, 2018

Problem 1

Maximize: $\sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs}$
Subject to:

$$
\begin{gathered}
f_{sv_1} \leq 16 \\
f_{sv_2} \leq 13 \\
f_{v_1v_3} \leq 12 \\
f_{v_2v_1} \leq 4 \\
f_{v_2v_4} \leq 14 \\
f_{v_3v_2} \leq 9 \\
f_{v_3t} \leq 20 \\
f_{v_4v_3} \leq 7 \\
f_{v_4t} \leq 4 \\
f_{v_1v_3} = f_{sv_1} + f_{v_2v_1} \\
f_{v_2v_1} + f_{v_2v_4} = f_{sv_2} + f_{v_3v_2} \\
f_{v_3v_2} + f_{v_3t} = f_{v_1v_3} + f_{v_4v_3} \\
f_{v_4t} + f_{v_4v_3} = f_{v_2v_4} \\
f_{uv} \geq 0 \ \forall u, v \in V
\end{gathered}
$$

Problem 2

Step 1: Convert the linear program to slack form:
Maximize $z = 5x_1 + 4x_2 + 3x_3$

Subject to:

$$
\begin{gathered}
x_4 = 5 - 2x_1 - 3x_2 - x_3 \\
x_5 = 11 - 4x_1 - x_2 - 2x_3 \\
x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\
x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{gathered}
$$

Step 2 Pivot on x_1:

For each constraint, when x_1 is isolated, the other terms are all negative aside from the constant. Therefore, the tightest constraint is the one with the smallest constant, which corresponds to row 1: $x_4 = 5 - 2x_1 - 3x_2 - x_3$. Solving for x_1 in this constraint provides $x_1 = \frac{5}{2} - \frac{x_4}{2} - \frac{3x_2}{2} - \frac{x_3}{2}$. We now substitute this back into the constraints:

$$
\begin{gathered}
z = 12.5 - 2.5x_4 - 3.5x_2 + 0.5x_3 \\
x_1 = 2.5 - 0.5x_4 - 1.5x_2 - 0.5x_3 \\
x_5 = 1 + 2x_4 + 5x_2 \\
x_6 = 0.5 + 1.5x_4 + 0.5x_2 - 0.5x_3 \\
x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{gathered}
$$

Again, now the non-basic variable with the largest coefficient in the objective function is x_3. The coefficients on x_3 are equivalent in the remaining constraints for which they are not 0. Therefore, the row with the lowest constant is the next constraint, which corresponds to the last row. Isolating x_3 yields the following: $x_3 = 1 - 2x_6 + 3x_4 + x_2$. Substitution yields the following constraints:

$$
\begin{gathered}
z = 13 - x_4 - 3x_2 - x_6 \\
x_1 = 2 - 2x_4 - 2x_2 + x_6 \\
x_5 = 1 + 2x_4 + 5x_2 \\
x_3 = 1 - 2x_6 + 3x_4 + x_2 \\
x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{gathered}
$$

All the coefficients on the basic variables are negative in the objective function, so the basic feasible solution is the optimal solution. Setting the non-basic variables in the objective function to 0 yields the optimal solution:

$$
\begin{aligned}
z &= 13 - 0 - 3*0 - 0 \\
x_1 &= 2 - 2*0 - 2*0 + 0 \\
x_5 &= 1 + 2*0 + 5*0 \\
x_3 &= 1 - 2*0 + 3*0 + 0 \\
&x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}
$$

The optimal solution in this modified objective function is $z = 13$ with $(x_1, x_2, ..., x_6) = (2, 0, 1, 0, 1, 0)$. Returning to the original objective function, the optimal solution is $z = 5*2 + 4*0 + 3*1 = 13$, with $(x_1, x_2, x_3) = (2, 0, 1)$.

Problem 3

Step 1: Convert the linear program to a maximization problem in slack form. Note that we change the "greater than or equal to" inequality to a "less than or equal to" by multiplying by -1:

Maximize $z = -3x_1 - 4x_2$

Subject to:

$$
\begin{aligned}
x_3 &= 600 - 2x_1 - x_2 \\
x_4 &= 225 - x_1 - x_2 \\
x_5 &= 1000 - 5x_1 - 4x_2 \\
x_6 &= -150 + x_1 + 2x_2 \\
&x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}
$$

Step 2: The basic solution is not feasible, so we must convert the constraints to a form that yields a basic feasible solution. Instead, solve the dual of the problem. Express the objective function and the constraints as a matrix and find its transpose:

$$
\begin{bmatrix}
-2 & -1 & . & -600 \\
-1 & -1 & . & -225 \\
-5 & -4 & . & -1000 \\
1 & 2 & . & 150 \\
. & . & . & . \\
3 & 4 & . & 0
\end{bmatrix}
$$

the transpose is:

$$\begin{bmatrix} -2 & -1 & -5 & 1 & . & 3 \\ -1 & -1 & -4 & 2 & . & 4 \\ . & . & . & . & . & . \\ -600 & -225 & -1000 & 150 & . & 0 \end{bmatrix}$$

the new objective function is Maximize $z = -600y_1 - 255y_2 - 1000y_3 + 150y_4$
Subject to:

$$\begin{aligned} -2y_1 - y_2 - 5y_3 + y_4 &\leq 3 \\ -y_1 - y_2 - 4y_3 + 2y_4 &\leq 4 \\ y_1, y_2, y_3, y_4 &\geq 0 \end{aligned}$$

Step 3: Convert the dual to slack form:
Maximize $z = -600y_1 - 255y_2 - 1000y_3 + 150y_4$

$$\begin{aligned} y_5 &= 3 + 2y_1 + y_2 + 5y_3 - y_4 \\ y_6 &= 4 + y_1 + y_2 + 4y_3 - 2y_4 \\ y_1, y_2, y_3, y_4 &\geq 0 \end{aligned}$$

Step 4: Pivot on y_4 with constraint in row 2. This yields the equation: $y_4 = 2 + 0.5y_1 + 0.5y_2 + 2y_3 - 0.5y_6$. Substitution yields the following problem:
Maximize $z = -525y_1 - 175y_2 - 700y_3 + 300 - 75y_6$

$$\begin{aligned} y_5 &= 1 + 1.5y_1 + 0.5y_2 + 3y_3 + 0.5y_6 \\ y_4 &= 2 + 0.5y_1 + 0.5y_2 + 2y_3 - 0.5y_6 \\ y_1, y_2, y_3, y_4 &\geq 0 \end{aligned}$$

All of the non-basic variables in the objective function now have negative coefficients, so the basic feasible solution is now the optimal solution. The optimal value is $z = 300$ with the values $(0, 0, 0, 2, 1, 0)$. The original values of x correspond to the slack variables in the dual, so the optimal values of $(x_1, x_2) = (y_5, y_6) = (1, 0)$.

Problem 4

a

$$\begin{aligned} A(1) &= 5 \\ A(i) &= 3i \\ A(-1) &= -1 \\ A(-i) &= -3i \end{aligned}$$

b

$$\begin{aligned} B(1) &= 8 \\ B(i) &= i + 7 \\ B(-1) &= 6 \\ B(-i) &= -i + 7 \end{aligned}$$

c

$$\begin{aligned} C(1) &= 40 \\ C(i) &= -3 + 21i \\ C(-1) &= -6 \\ C(-i) &= -3 - 21i \end{aligned}$$

d

We must interpolate by taking the inverse DFT of these point-wise multiplications. The inverse DFT is given by:

$$c(k) = \frac{1}{N} \sum_{j=0}^{N-1} C(j) \omega_N^{-kj} \tag{1}$$

where k indexes the point-wise multiplications from (c) and ω_N^{nk} is the nk-th power of the N-th root of unity. This can be expressed as a matrix operation. Using the four coefficients available, and hence N=4, the coefficients of C can be written as the matrix product:

$$(\omega_4^{-0}, \omega_4^{-1}, \omega_4^{-2}, \omega_4^{-3}) = (1, -i, -1, i) \tag{2}$$

$$C = \frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -i & -1 & i\\ 1 & (-i)^2 & (-1)^2 & i^2\\ 1 & (-i)^3 & (-1)^3 & i^3\end{bmatrix}\begin{bmatrix}40\\ -3+21i\\ -6\\ -3-21i\end{bmatrix} = \begin{bmatrix}7\\ 22\\ 10\\ 1\end{bmatrix}$$

The coefficients of C are therefore $C(x) = x^3 + 10x^2 + 22x + 7$. Directly calculating the inverse DFT does not yield a $\Theta(n \log n)$ time. Perhaps you already see the computation can be optimized the same way as FFT. So to calculate C's coefficient in $\Theta(n \log n)$ time, one way you could do is to apply the same FFT algorithm to inverse DFT. And here is another simpler $\Theta(n \log n)$ method that is essentially equivalent to FFT. Observe that

$$C(1) + C(-1) = 2(b + d) = 34 \tag{3}$$

$$C(i) + C(-i) = 2(d - b) = -6 \tag{4}$$

By doing this we are dividing the original problem of size 4 into two subproblems with size 2 in linear time. We do the same for a and c.

$$C(1) - C(-1) = 2(a + c) = 46 \tag{5}$$

$$C(i) - C(-i) = 2i(c - a) = 42i \tag{6}$$

Apply the same division again on the four equations about a, c and b, d, we can further divide the currently two problems to four subproblems with size 1, which are directly solvable and the recurrence stops.