Optimization and Computational Biology (Spring 2008)
Brief Solutions to Homework 3

1. Let \(x_i, y_i, \) and \(t_i\) denote the number of units produced in regular time, over time, and burn-midnight-oil (BMO) time, respectively, in period \(i\), for \(i = 1, 2, 3\). Also, let \(s_i\) be the number of units in inventory at the end of period \(i\), for \(i = 0, 1, 2, 3\). Here, \(s_0\) denotes the starting inventory (225 units). The LP formulation is given below. For ease of notation, denote the demands for the three periods by \(d_1 = 5400\), \(d_2 = 6100\), and \(d_3 = 6000\).

\[
\begin{align*}
\text{min } & \quad z = 10 \sum_{i=1}^{3} x_i + 14 \sum_{i=1}^{3} y_i + 25 \sum_{i=1}^{3} t_i + 4 \sum_{i=1}^{3} s_i \quad \text{(total cost)} \\
\text{s.t.} & \quad (x_i + y_i + t_i) + s_{i-1} - d_i = s_i, \quad i = 1, 2, 3 \quad \text{(inv. balance for month } i) \\
& \quad x_1 \leq 5000 \quad \text{(period 1 reg time cap)} \\
& \quad x_2 \leq 5500 \quad \text{(period 2 reg time cap)} \\
& \quad x_3 \leq 3200 \quad \text{(period 3 reg time cap)} \\
& \quad y_1 \leq 1000 \quad \text{(period 1 overtime cap)} \\
& \quad y_2 \leq 1100 \quad \text{(period 2 overtime cap)} \\
& \quad y_3 \leq 950 \quad \text{(period 3 overtime cap)} \\
& \quad t_i \leq 500, \quad i = 1, 2, 3 \quad \text{(period } i \text{ BMO cap)} \\
& \quad s_0 = 225 \quad \text{(starting inv.)} \\
& \quad s_i \geq 200 \quad \text{for } i = 1, 2, 3 \quad \text{(period } i \text{ min inv.)} \\
& \quad x_i, y_i, t_i, s_i \geq 0 \quad \text{for } i = 1, 2, 3 \quad \text{(non-negativity)}
\end{align*}
\]

See course web page for the AMPL model and data files. The minimum total cost is $2081.35.

2. We can do row and column interchanges to transform \(I(A, B)\) into a “staircase” form (this transformation is not unique).

\[
I(A, B) = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

The corresponding restriction maps are \(A_3A_5A_1A_4A_2\) and \(B_1B_7B_8B_5B_2B_3B_6B_4\), with the edges evident from the transformed \(I(A, B)\) given above.

Notice that the intervals in each of the two sets \((A \text{ and } B)\) are non-overlapping within the sets themselves (i.e., \(A_i \cap A_j = \emptyset \) for \(i \neq j\)). Hence each vertex \((A_i \text{ or } B_j)\) will have at most two adjacent vertices \(v\) with \(\deg(v) \geq 2\). One could start working with \(L = \{v : \deg(v) \geq 2\}\), and locate an “end” – a vertex \(v\) which has only one adjacent vertex \(u \in L\). This vertex set \(\{v_1 = v, v_2 = u\}\) can be extended to other vertices in \(L\) connected to the vertices already included (you will identify all the vertices in the connected component of the graph containing \(v \text{ and } u\)). The order in which the vertices are sequentially included will define (part of the) desired ordering of \(A_i \text{ and } B_j\). You could then take care of single intervals and isolated pairs.

The algorithm outlined above can be modified so that it checks whether a given graph is an interval graph or not. This algorithm requires linear time and storage (in \(m \text{ and } n\), the number of \(A_i\)’s and \(B_j\)’s).
3. Notice that each interval in the double digest $C_k$ overlaps with exactly one interval from the $A$ digest, and exactly one from the $B$ digest (say, $A_i$ and $B_j$ respectively). Thus, each column of $I(A, C)$ and $I(B, C)$ (equivalently, each row of $[I(B, C)^T]$) has exactly one entry equal to 1, and the remaining entries are all zeros. Further, $A_i \cap C_k \neq \emptyset \land B_j \cap C_k \neq \emptyset \Rightarrow A_i \cap B_j \neq \emptyset$. The result in question follows from these arguments.

4. In order to generalize the result for DDP to higher order digests, one should realize that there is nothing essential about the matrix product form – it makes more sense to write the result as a sum-product of two vectors (which has the nice representation as a matrix product when there are only two vectors):

$$I(A, B)_{ij} = \sum_{k=1}^{l} I(A, C)_{ik} \cdot I(B, C)_{jk}.$$ 

Let $A_1, \ldots, A_r$ denote the digests with the $r$ individual enzymes, having $n_1, \ldots, n_r$ components, respectively. Also, let $X$ denote the $r$-tuple digest, with $l$ components. The intersection result in this case can be stated as follows.

$$A_{1_{i_1}} \cap \cdots \cap A_{r_{i_r}} \neq \emptyset \iff A_{1_{i_1}} \cap X_k \neq \emptyset \land \cdots \land A_{r_{i_r}} \cap X_k \neq \emptyset \text{ for some } k \in \{1, \ldots, l\}.$$ 

Thus, we can extend the sum-product result to the case of the $r$-digest problem:

$$I(A_1, \ldots, A_r)_{i_1, \ldots, i_r} = \sum_{k=1}^{l} \prod_{j=1}^{r} I(A_j, X)_{i_j, k}, \text{ for } 1 \leq i_j \leq n_j, j = 1, \ldots, r.$$ 

5. See course web page for AMPL model and data files.