CSC373 Worksheet 3 Solution

July 28, 2020

1. **Notes:**

- Dynamic Programming
  - Is applied to optimization problems
  - Applies when the subproblems overlap
  - Uses the following sequence of steps
    1. Characterize the structure of an optimal solution
    2. Recursively define the value of an optimal solution
    3. Construct an optimal solution from computed information

- Matrix-chain Multiplication
  - Is an optimization problem solved using dynamic programming
  - Goal is to find matrix parenthesis with fewest number of operations

    Example:

    Given chain of matrices $< A, B, C >$, it's fully parenthesized product is:

    * $(AB)C$ needs $(10 \times 30 \times 5) + (10 \times 5 \times 60) = 1500 + 3000 = 4500$ operations
    * $A(BC)$ needs $(30 \times 5 \times 60) + (10 \times 30 \times 60) = 27000$ operations

    Thus, $(AB)C$ performs more efficiently than $A(BC)$.

  - Is stated as: given a chain $< A_1, A_2, ..., A_n >$ of n matrices, where for $i = 1, 2, ..., n$ matrix A_i has dimension $p_{i-1} \times p_i$, fully parenthesize the product $A_1A_2...A_n$ in a way that minimizes the number of scalar multiplications.
  - Steps

1. **Check is the problem has Optimal Substructure**

   Let us adopt the notation $A_{i...j}$ where $i \leq j$, for the matrix that results from evaluating the product $A_i A_{i+1} ... A_j$.

   Assume the solution has the following parentheses:

   $$(A_{i...k})(A_{k+1...j})$$

   If there is a better way to multiply $(A_{i...k})$, then we would have a more optimal solution.

   This would be a contradiction, as we already stated that we have the optimal solution for $A_{i...j}$.

   Therefore, this problem has optimal substructure.

2. **Find the Recursive Solution**

   Let $M[i, j]$ be the cost of multiplying matrices from A_i to A_j

   We want to find out at which $'k'$ returns the fewest number of multiplications, or the minimum number of M.

   The recursive formula for the cost of multiplying from A_i to A_j is

   $$M[i,j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \leq k \leq j} M[i,k] + M[k+1,j] + p_{i-1} p_k p_j & \text{if } i < j \end{cases} \tag{1}$$

3. **Computing the Estimated Cost**

   ∗ Steps

   1) Fill the table for $i = j$
   2) Fill the table for $i < j$ with a spread of 1
   3) Repeat 2 with the increased value of spread

   Example:

   Given
   $< A_1, A_2, A_3, A_4, A_5 >$

   where
   ∗ $A_1 \rightarrow 4 \times 10$
   ∗ $A_2 \rightarrow 10 \times 3$
   ∗ $A_3 \rightarrow 3 \times 12$

* $A_4 \to 12 \times 20$
* $A_5 \to 20 \times 7$

we have:

1) Fill the table for $i = j$

1) i = j

i \ j	1	2	3	4	5
1	0				
2	x	0			
3	x	x	0		
4	x	x	x	0	
5	x	x	x	x	0

$$M[i,j] = \begin{cases} \boxed{0 \qquad\qquad \text{if } i = j} \\ \min_{i \le k < j} M[i,k] + M[k+1,j] + p_{i-1}p_kp_j & \text{if } i < j \end{cases}$$

2) Fill the table for $i < j$ with a spread of 1

2) (i = 1, j = 2), (i = 2, j = 3), (i = 3, j = 4), (i = 4, j = 5)

i \ j	1	2	3	4	5
1	0				
2	x	0			
3	x	x	0		
4	x	x	x	0	
5	x	x	x	x	0

since

* $i = 1, j = 2$

$$\begin{aligned} M[1,2] &= \min_{1 \le k \le 2} (M[1,1] + M[1,2] + p_{i-1}p_kp_j) && (2) \\ &= \min_{1 \le k \le 2} (0 + p_0p_1p_2) && (3) \\ &= \min_{1 \le k \le 2} (0 + 4 \cdot 10 \cdot 3) && (4) \\ &= 120 && (5) \end{aligned}$$

where $p_0 = 3$ is from the dimension 3×10 of A_1, $p_k = 10$ is from the dimension of 3×10 of A_1.

3) Repeat 2 with the increased value of spread

4. **Constructing the Optimal Solution**

References:

1)