Department of Computer Science and Engineering

PES UNIVERSITY

UE19CS251: Design and Analysis of Algorithms (4-0-0-4-4)

Exhaustive Search: Assignment Problem

Dr. Shylaja S S

Assignment Problem

There are n people who need to be assigned to execute n jobs, one person per job. (That is, each person is assigned to exactly one job and each job is assigned to exactly one person.) The cost that would accrue if the i^{th} person is assigned to the j^{th} job is a known quantity C[i, j] for each pair i, j = 1, 2, ... , n. The problem is to find an assignment with the minimum total cost.

A small instance of this problem follows, with the table entries representing the assignment costs C[i, j]:

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

It is easy to see that an instance of the assignment problem is completely specified by its cost matrix C. In terms of this matrix, the problem calls for a selection of one element in each row of the matrix so that all selected elements are in different columns and the total sum of the selected elements is the smallest possible. Note that no obvious strategy for finding a solution works here. For example, we cannot select the smallest element in each row because the smallest elements may happen to be in the same column. In fact, the smallest element in the entire matrix need not be a component of an optimal solution. Thus, opting for the exhaustive search may appear as an unavoidable evil.

We can describe feasible solutions to the assignment problem as n-tuples $\langle j_1, \ldots, j_n \rangle$ in which the i^{th} component, i = 1, ... , n, indicates the column of the element selected in the i^{th} row (i.e., the job number assigned to the i^{th} person). For example, for the cost matrix above, $\langle 2, 3, 4, 1 \rangle$ indicates a feasible assignment of Person 1 to Job 2, Person 2 to Job 3, Person 3 to Job 4, and Person 4 to Job 1. The requirements of the assignment problem imply that there is a one-to-one correspondence between feasible assignments and permutations of the first n integers. Therefore, the exhaustive-search approach to the assignment problem would require generating all the permutations of integers 1, 2, ..., n, computing the total cost of each assignment by summing up the corresponding elements of the cost matrix, and finally selecting the one with the smallest sum. A few first iterations of applying this algorithm to the instance given above are shown in Fig. 1; you may complete the remaining.

$$C = \begin{bmatrix} 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix}$$

$<1, 2, 3, 4>$	cost = 9 + 4 + 1 + 4 = 18	
$<1, 2, 4, 3>$	cost = 9 + 4 + 8 + 9 = 30	
$<1, 3, 2, 4>$	cost = 9 + 3 + 8 + 4 = 24	
$<1, 3, 4, 2>$	cost = 9 + 3 + 8 + 6 = 26	etc.
$<1, 4, 2, 3>$	cost = 9 + 7 + 8 + 9 = 33	
$<1, 4, 3, 2>$	cost = 9 + 7 + 1 + 6 = 23	

Fig. 1: First few iterations of solving a small instance of the assignment problem by exhaustive search

Since the number of permutations to be considered for the general case of the assignment problem is n!, exhaustive search is impractical for all but very small instances of the problem. Fortunately, there is a much more efficient algorithm for this problem called the Hungarian method after the Hungarian mathematicians Konig and Egervary whose work underlies the method.