



A T M E
College of Engineering



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

Design and analysis of algorithms

BCS401



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING - AI & ML

Module-5

LIMITATIONS OF ALGORITHMIC POWER: Decision Trees, P, NP, and NP-Complete Problems.

COPING WITH LIMITATIONS OF ALGORITHMIC POWER: Backtracking (n-Queens problem, Subset-sum problem), Branch-and-Bound (Knapsack problem), Approximation algorithms for NP-Hard problems (Knapsack problem).

Chapter 11 (Section 11.2, 11.3), Chapter 12 (Sections 12.1,12.2,12.3)

Backtracking

The problem is to place n queens on an $n \times n$ chessboard so that no two queens attack each other by being in the same row or in the same column or on the same diagonal.

For $n = 1$, the problem has a trivial solution, and it is easy to see that there is no solution for $n = 2$ and $n = 3$.

So let us consider the four-queens problem and solve it by the backtracking technique. Since each of the four queens has to be placed in its own row, all we need to do is to assign a column for each queen on the board presented in Figure 12.1.

Subset-Sum Problem

ALGORITHM *Backtrack*($X[1..i]$)

//Gives a template of a generic backtracking algorithm

//Input: $X[1..i]$ specifies first i promising components of a solution

//Output: All the tuples representing the problem's solutions

if $X[1..i]$ is a solution **write** $X[1..i]$

else //see Problem 9 in this section's exercises

for each element $x \in S_{i+1}$ consistent with $X[1..i]$ and the constraints **do**

$X[i + 1] \leftarrow x$

Backtrack($X[1..i + 1]$)