Limits of Computational Problem Solving

Traveling Salesman problem

The Traveling Salesman problem is a classic computational problem in computer science. The problem is to find the shortest route of travel for a salesman needing to visit a given set of cities. In a brute force approach, the lengths of all possible routes would be calculated and compared to find the shortest one. For ten cities, the number of possible routes is 10! (10 factorial), or over three and a half million (3,628,800). For twenty cities, the number of possible routes is 20! or over two and a half quintillion (2,432,902,008,176,640,000).

If we assume that a computer could compute the lengths of one million routes per second, it would take over 77,000 years to find the shortest route for twenty cities by this approach. For 50 cities, the number of possible routes is over 10^{64}. In this case, it would take more time to solve than the age of the universe!

Game of Chess

A similar problem exists for the game of chess. A brute force approach for a chess-playing program would be to "look ahead" to all the eventual outcomes of every move that can be made in deciding each next move. There are approximately 10^{120} possible chess games that can be played. This is related to the average number of look-ahead steps needed for deciding each move. How big is this number? There are approximately 10^{80} atoms in the observable universe, and an estimated 3 X 10^{90} grains of sand to fill the universe solid. Thus, ***there are more possible chess games that can be played than grains of sand to fill the universe solid!***

For problems such as this and the Traveling Salesman problem in which a brute-force approach is impractical to use, more efficient problem-solving methods must be discovered that find either an exact or an approximate solution to the problem.

Any algorithm that correctly solves a given problem must solve the problem in a reasonable amount of time, otherwise it is of limited practical use.