**Week 2: Mini Literature Review**

*Submitted by: Sandip Sahani, Iksha Gurung, Milan Thapa*

**Problem Description:**

The “8 Queen problem” is a problem of placing 8 queens in a 8X8 chess board such that no two Queens can attack each other. It is one among a series of n other such problems where n queens are to be placed in a nXn chess board such that no two queens could attack each other. The solution for this problem is to place each queens in such a way that the column, row or the diagonals of the box where the queens are placed intersect. There are only 92 such combinations possible amongst 4,426,165,368 combinations of placing the queens.

**Determining the solution:**

The 8 queen problem may be very expensive in terms of computing as one has to try out 4,426,165,368 combinations. Of course one does not use a program that takes such high computing into consideration. There are different ways the problem can be simplified, in other words the computing cost be deduced.

One of the many ways of deducing the computing cost and determining the solution is to use the Backtracking algorithm. For the given problem (8 queen problem) the backtracking depth-first search algorithm constructs the search tree by considering one row of the board at a time, eliminating most non-solution board positions at a very early stage in their construction. Because it rejects rook and diagonal attacks even on incomplete boards, it examines only 15,720 possible queen placements. A further improvement which examines only 5,508 possible queen placements is to combine the permutation based method with the early pruning method: the permutations are generated depth-first, and the search space is cut back if the partial permutation produces a diagonal attack.

**Pseudo Code for the 8 queen problem using backtracking depth-first search algorithm:**

function QUEEN-SEARCH(queen[1..n])

begin

repeat

//generate a random permutation of queen[1] to queen[n].

forall i,j ;where queen[i] or queen[j]is attacked do

if (swap(queen[i],queen[j]))reduces collision

then perform-swap(queen[i],queen[j])

until no collision

end

In words the instructions to be given are:

1. Tests whether there is no attack
2. If found no-attack, return it
3. Else for each choice that can be made
4. Make that choice
5. Recursion
6. If recursion returns a no-attack, return it
7. If no choices remain, return failure

The solution of the problem can be determined using the backtracking algorithm in the using next instruction set. The instruction set are:

– Identify data structures to solve the problem

-First pass: Define the chessboard to be an 8 × 8 array

-Second pass: Since each queen is in a different row, define the chessboard solution to be an 8-tuple (x1, . . . , x8),where xi is the column for ith queen

– Identify explicit constraints

-Explicit constraints using 8-tuple formulation are Si = {1, 2, 3, 4, 5, 6, 7, 8}, 1 i 8

-Solution space of 88 8-tuples

– Identify implicit constraints

-No two xi can be the same, or all the queens must be in different columns

-All solutions are permutations of the 8-tuple (1, 2, 3, 4, 5, 6, 7, 8)

- Reduces the size of solution space from 88 to 8! tuples

-No two queens can be on the same diagonal

– The solution above is expressed as an 8-tuple as 4, 6, 8, 2, 7, 1, 3, 5.

**References:**

1. <http://en.wikipedia.org/wiki/Eight_queens_puzzle> Date:May 4, 2012,Time:8:40pm
2. <http://www.scribd.com/doc/1017770/N-Queen-Problem> Date:May 4, 2012,Time:8:38pm
3. <http://bridges.canterbury.ac.nz/features/eight.html> Date:May 4, 2012,Time:8:35pm