**COMP 3710 Applied Artificial Intelligence**

**Seminar/Lab 4.**

**Local search and CSPs**

1. **Objectives**
* Use of local search for *n*-queens problem.
* Use of ‘Most-Constrained Variable First’ heuristic for a constraints satisfaction problem.
1. **Exercise – Local search for 5-queens problem**
* Initial board

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 |  |  |  |  | Image result for queen  symbol |
| 3 |  |  |  |  |  |
| 2 | Image result for queen  symbol |  | Image result for queen  symbol |  |  |
| 1 |  | Image result for queen  symbol |  |  |  |
| 0 |  |  |  | Image result for queen  symbol |  |
|  | 0 | 1 | 2 | 3 | 4 |

* Show how local search can be used to find a solution. At each step, you need to show the evaluation of board. (Note that local search can be struck to a local optimum, i.e., not always solution.)

|  |  |  |
| --- | --- | --- |
| Board | The number of conflicts | Selected column and value |
| [2, 1, 2, 0, 4] | 5 |  |
|  |  |  |
| … |  |  |

1. **Exercise – CSP**
* Trace the operation of Most-Constrained Variables First (MCVF) for the 5-queens problem.
	+ MCVF:
		- Most constrained variable: You can select a variable that has the least number of choices.
	+ List **at least 6 steps**, or till a solution is found.
	+ You need to show how next variables are selected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Board | # of constrainsfor column-0 | # of constrains for column-1 | # of constrains for column-2 | # of constrains for column-3 | # of constrains for column-4 | Selection |
| [-1, -1, -1, -1, -1] | 0 | 0 | 0 | 0 | 0 | 0:0 |
| [ 0, -1, -1, -1, -1] |  | 2 | 2 | 2 | 2 | 1:3 |
| … |  |  |  |  |  |  |
| … |  |  |  |  |  |  |

1. **Exercise – Local search for *n*-queens problem**

For limited number of trials,

 Initialize columns; // columns is an array for the positions of queen with the restriction

 // that each column has one queue. E.g., columns[2] is the row

 // number of the queen in column-2.

 Local search with columns;

 If columns is a goal,

 break;

If a goal is found,

 display the board;

1. **Exercises for local search algorithm for *n*-queens problem**
* Functions
	+ initialize(columns)
	+ countConflicts(columns, col) – count the number of conflicts with the queen in the specified column
	+ isGoal(columns)
	+ improve(columns, col) – improve the board over the specified column
	+ localSearch(columns)
* Complete the posted exercise program.
1. **Local search algorithm for *n*-queens problem**
* Complete the posted program that solves *n*-queens problem using local search. You can use the functions in the above exercise program.
1. **Assignment**
	1. You will be given roughly 1 assignment or 2 assignments every week to help you understand all the topics in the lectures. The title of the email should include your name, id, and COMP 3710.
	2. Submission
* A document for the two exercises in 2) and 3). Any document that include hand drawn images will NOT be accepted.
* The program in 6)
	+ Due:
		- 6:00 pm, January 29, 2019 – with bonus 10%
		- 6:00 pm, January 30, 2019 – with the full marks
		- 6:00 pm, January 31, 2019 – with penalty 5%
		- 6:00 pm, February 1, 2019 – with penalty 10%
* Total marks: 5 + 10
	+ - Any syntax error will get zero.
		- No completion will get zero.
	1. Any late submission will NOT be accepted.
	2. You should NOT use any assignments submitted in the previous semesters.