**COMP 3710 Applied Artificial Intelligence**

**Seminar/Lab 4.**

**Genetic Algorithms**

1. **Objectives**

* Getting familiar with genetic algorithms with the 8-queens problem

1. **Genetic algorithm for the 8-queens problem**

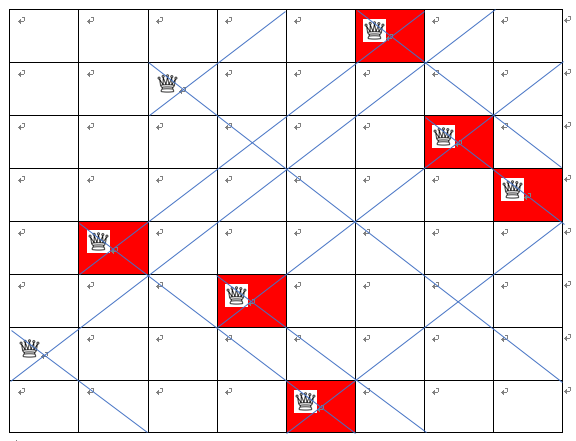
* Trace the operation of the genetic algorithm with the example of the 8-queens problem.
  + You should generate next populations **at least 3 times**.
  + Here is the initial population.
    - [2 4 7 3 1 8 6 5], [1 2 3 4 5 6 7 8], [2 4 1 4 5 1 2 4], [2 3 4 5 2 3 1 3]
  + Fitness function
    - Number of non-attacking pairs of queens = (28 – the number of conflicts)
    - The fitness of an individual = the number of non-attacking pairs of queens in the individual / the total number of non-attacking pairs of queens in all the individuals \* 100
  + Selection: roulette selection
    - E.g., 31%, 29%, 26%, 14%
    - From above, construct a list – [0, 31, 60 (=31 + 29), 86 (=60 + 26), 100 (=86 + 14)].
  + Run a random number generator to get a random number, *r*, in [0, 100]. You can use <http://cs.tru.ca/~mlee/comp3710/Software/random.html> to generate random numbers.
    - How to select with a random number r?
      * With the above example, if *r* <= 31, then the first one is selected
      * else if *r* <= 60, then the second one is selected
      * else if *r* <= 86, then the third one is selected
      * else, the last one is selected.
  + Crossover
    - Middle point cross over
  + Mutation:
    - Mutation rate =20% (Actually it is extremely high in real applications. This high mutation rate would make the system unstable. But just for practice here.)
    - How to do it?
      * For each new individual, decide random number *m* in [0, 100].
      * If *m* <= 20, then
        + Select one variable randomly, i.e., *v* in [1, 8].
        + Select one value randomly, i.e., *p* in [1, 8]
        + Update the value of *v* with *p*.
  + You can use <http://cs.tru.ca/~mlee/comp3710/Software/random.html> to generate random numbers.
  + For each generation, you may use the following two tables.

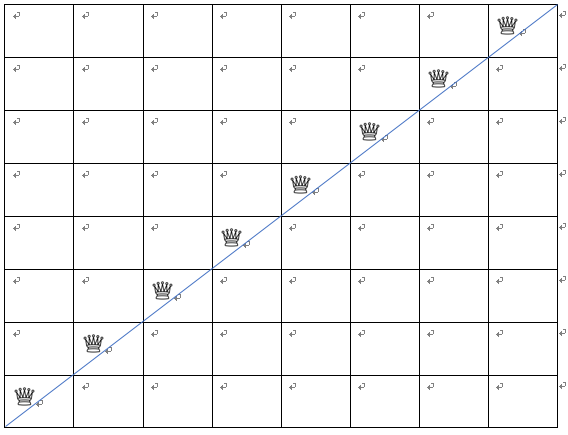
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Individuals | | Fitness values | | Fitness ratios (%) | | |
| [2 4 7 3 1 8 6 5] | |  | |  | | |
| [1 2 3 4 5 6 7 8] | |  | |  | | |
| [2 4 1 4 5 1 2 4] | |  | |  | | |
| [2 3 4 5 2 3 1 3] | |  | |  | | |
| *r* | Selected individuals | | New individuals | | *m* | *v* | | *p* | Mutated individuals |
|  |  | |  | |  |  | |  |  |
|  |  | |  | |  |  | |  |  |
|  |  | |  | |  |  | |  |  |
|  |  | |  | |  |  | |  |  |

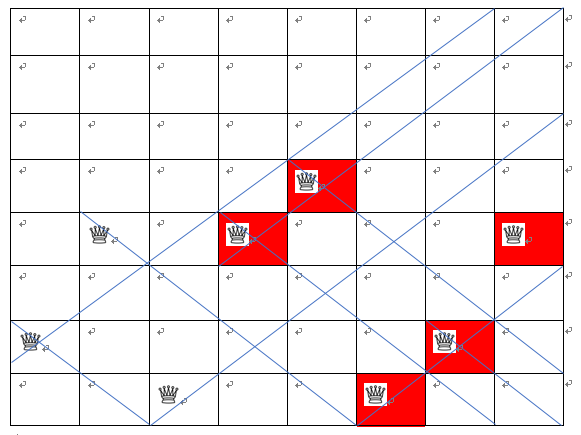
* Show the 2nd generation and the 3rd generation with the above tables.
* This exercise of genetic algorithm is very crucial to the success of the next week work.

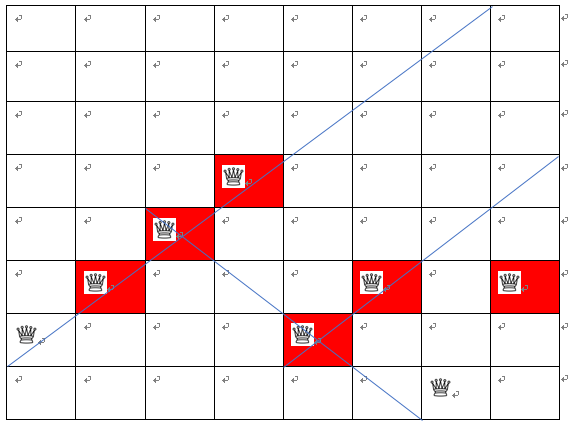
1. **Assignment**
   1. You will be given roughly 1 assignment or 2 assignments every week to help you understand all the topics in the lectures.
   2. Submission

* Submit a document for 2) by email.
  + The title of the email should include your name, id, and COMP 3710.
  + Due:
    - 6:00 pm, October 11, 2017
    - Any late submission will NOT be accepted. Even 1 second according to TRU email system.
* Total marks: 10.









r = 47.392886263782195

[2 4 1 4 5 1 2 4]

[2 4 7 3 1 8 6 5]

[2 4 1 4 5 1 2 4]

[2 3 4 5 2 3 1 3]

r = 32.32696917027824 <40

[2,4,7,3,1,8,6,5] [2,4,7,3,5,1,2,4]

[2,4,1,4,5,1,2,4] [2,4,1,4,1,8,6,5]

[2,4,7,3,1,8,6,5] [2,4,7,3,2,3,1,3]

[2,3,4,5,2,3,1,3] [2,3,4,5,1,8,6,5]

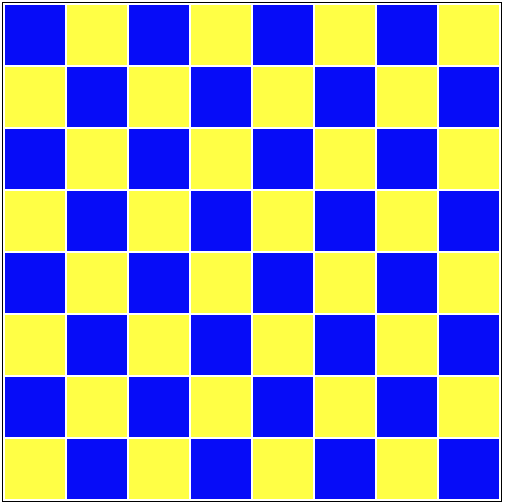
r = 47.392886263782195

[2 4 1 4 5 1 2 4]

[2 4 7 3 1 8 6 5]

[2 4 1 4 5 1 2 4]

[2 3 4 5 2 3 1 3]



R= 51.769515533005794

[2,4,1,4,1,8,6,5]

[2,4,7,3,5,1,2,4]

[2,4,7,3,2,7,1,3]

[2,4,7,3,5,1,2,4]

[2,4,7,3,1,8,6,5]

[2,4,7,3,5,1,2,4]

[2,4,7,3,1,8,6,5]

[2,4,8,3,2,7,1,3]