**COMP 3710 Applied Artificial Intelligence**

**Seminar/Lab 5.**

**Propositional logic, and genetic algorithm for TSP**

1. **Objectives**

* Use of logical equivalences
* Implementation of a genetic algorithm to solve TSP

1. (2 marks) Which of the followings are correct? You need to prove or disprove using truth tables.
   1. (*A* ∧ *B*) → *C* ≡ (*A* → *C*) ∨ (*B* → *C*)
   2. (*C* ∨ (~*A* ∧ ~*B*)) ≡ ((*A* → *C*) ∧ (*B* → *C*))
2. (2 marks) Simplify the followings as much as possible.
   1. ~(~*A* ∧ *B*) ∧ (*A* ∨ *B*)
   2. ~((~*A* ∧ *B*) ∨ (~*A* ∧ ~*B*))
3. **How to implement a genetic algorithm for TSP**

// <script src=’//cs.tru.ca/~mlee/comp3710/Software/ga\_tsp.js’></script>

var NO\_CITIES = ...

var POPULATION\_SIZE = ...

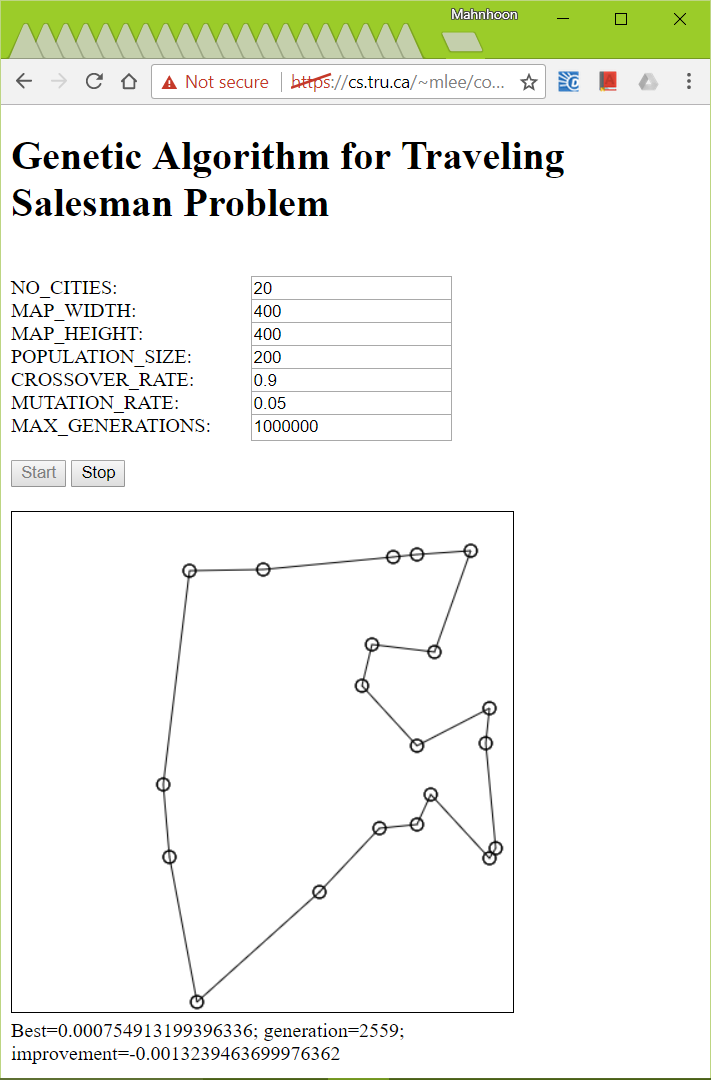
var CROSSOVER\_RATE = ...

var MUTATION\_RATE = ...

var MAX\_GENERATIONS = ...

var canvas = ...

var ctx = ...

var cities = initial\_cities(...); // City numbers: 0, 1, ...

// **City locations**: cities[i].x; cities[i].y

var population = get\_initial\_population(POPULATION\_SIZE, cities);

// Each individual is an array of all city numbers.

// population[i]: [2, 5, 12, 56, 22, ...]

...

function run() // This function is invoked every 5 ms by a timer.

{

// You need to implement this function.

population = **get\_next\_population**(population, cities);

// You need to implement this function.

var the\_best\_individual = **get\_best\_individual**(population, cities);

draw\_individual(the\_best\_individual, cities, ctx, canvas);

if (!(++count < MAX\_GENERATIONS))

clearInterval(timer);

}

The functions that you need to implement:

* + function **get\_next\_population**(current\_population, cities)

{

var next\_population = [];

... // Let’s discuss this part again.

return next\_population;

}

* + function **get\_best\_individual**(population, cities)

{

var the\_best\_individual = [];

... // Let’s discuss this part again.

return the\_best\_individual;

}

The functions in the library:

* + get\_initial\_population(population\_size, cities)
  + draw\_individual(individual, cities, ctx, canvas)
  + get\_distance\_between\_two\_cities(city\_a, city\_b, cities)

When your implementation is correct, TSP with 20-30 cities will be solved quickly.

You can start with

* w5\_ga\_tsp\_student.html

1. **Exercise**

* Fitness evaluation (Idea 3 at Slide 29):

var lengths = [];

for (var i = 0; i < POPULATION\_SIZE; i++) {

lengths[i] = get\_tour\_length (???, cities);

}

var fitness\_values = [];

for (var i = 0; i < POPULATION\_SIZE; i++) {

fitness\_values[i] = 1 / ???;

}

var sum = 0;

for (var i = 0; i < POPULATION\_SIZE; i++)

sum += ???;

var fitness\_ratios = [];

for (var i = 0; i < POPULATION\_SIZE; i++)

fitness\_ratios[i] = ??? / ???; // The sum of fitness\_ratios[i]’s becomes 1.

* Roulette selection:

var r = Math.???(); // r is in [0, 1)

var selected\_individual\_index;

var sum = 0;

for (var i = 0; i < POPULATION\_SIZE; i++) {

sum += ???;

if (r <= ???) {

selected\_inidividual\_index = i;

break;

}

}

* Crossover (Slide 32)

Select two parent individuals;

Generate a random number;

If it is within CROSSOVER\_RATE,

Decide the crossover point; // [1, individual\_size); Math.floor(Math.random() \* ???) + 1

Apply crossover to get the two offsprings;

Put them into the next population pool;

Else

???

Do the above steps multiple times so that the next population pool becomes full.

* Mutation (Slide 33)

For each individual in the next population pool,

Generate a random number;

If it is within MUTATAION\_RATE,

Decide two cities, a and b; // Let’s assume a < b

Apply the mutation;

1. **Submission**

* You need to submit a document file for 2) and 3), not images of hand-written answers.
  + Total marks: 4
  + Due: 6:00 PM, October 13, 2017
* You need to submit the html file for 4).
  + Total marks: 16
    - Any syntax error: 0 mark
    - No completion: 0 mark
    - Generally no partial marks for any code that cannot find an optimal solution. You really need to complete this programming.
  + Due: 6:00 PM, October 18, 2017
* No late submission will be accepted.