CS301 FS2007 Exam 2 Key

This is a closed-book, closed-notes exam. The only items you are allowed to use are writing implements. Mark each sheet of paper you use with your name and the string “cs301fs2007 exam2”. If you are caught cheating, you will receive a zero grade for this exam. The max number of points per question is indicated in square brackets after each question. The sum of the max points for all the questions is 28, but note that the max exam score will be capped at 27 (i.e., there is one bonus point but you can’t score more than 100%). While this exam has been designed to be doable in 45 minutes or less, you have up to 60 minutes to complete this exam. Keep your answers clear and concise while complete. Good luck!

Multiple Choice Questions - Circle the letter indicating your choice

1. Recombination has the potential to produce an offspring which is different than its parents in its: [1]
   (a) genotype
   (b) number of chromosomes
   (c) alleles
   (d) all of the above

2. The Baldwin Effect is: [1]
   (a) improved EA performance obtained by applying local search to phenotypes prior to fitness calculation
   (b) improved EA performance obtained by combining local search with Lamarckian evolution
   (c) improved EA performance obtained by combining local search with Darwinian evolution
   (d) none of the above

3. Countermeasures to Bloat in Genetic Programming include: [1]
   (a) reducing the number of alleles to prevent disproportional tree growth
   (b) reducing parsimony pressure to penalize the fitness of large chromosomes
   (c) increasing mutation rate to maintain genetic diversity
   (d) none of the above

4. Learning Classifier Systems are technically speaking: [1]
   (a) a type of Evolutionary Algorithm
   (b) a type of Reinforcement Learning System
   (c) a type of Condition-Action Rule-Based System
   (d) both of the last types

Regular Questions

5. Explain briefly the difference between “parameter tuning” and “parameter control”. [4]
   
   Parameter tuning refers to the a priori optimization of fixed strategy parameters, while parameter control refers to the on-the-fly optimization of dynamic strategy parameters.


   It refers to the optimal ratio of successful versus total mutations in Evolution Strategies where mutation step size is increased if the ratio is greater than $\frac{1}{5}$ and decreased if the ratio is smaller than $\frac{1}{5}$. 
7. Describe briefly the characteristics of Genetic Programming which set it apart from other types of Evolutionary Algorithms. [5]

GP is a class of EAs originally aimed at the automatic generation of computer programs, typically represented by parse trees (more in general, GP can be used for the automatic generation of symbolic expressions representing computer programs, mathematical equations, etc.). Each individual encodes a complete computer program in some programming language. To evaluate an individual the program encoded in its genes is executed and the fitness function assigns some degree of success to its execution. Special genetic operators that work on parse trees are required. Special care has to be taken to satisfy the constraint of producing syntactically correct computer programs.

8. What is the closure property in Genetic Programming? [2]

Each function in the function set can take any valid expression as an argument.


The rules are grouped by advocated action and the action advocated by the group with the highest average rule fitness is selected and those rules form the action set.

10. Describe concisely the difference between the Pitt approach and the Michigan approach in Learning Classifier Systems. [2]

In the Pitt approach each individual represents a complete rule set, while in the Michigan approach each individual represents a single rule and the entire population represents the complete rule set.
11. Show how one can configure an EA to be functionally equivalent to Simulated Annealing (SA), or, if you believe it is inherently impossible, explain why. [5]

For your convenience, here is some SA pseudo-code:

\[
\text{function SimulatedAnnealing(problem, schedule) returns a solution state}
\]
\[
\text{current} \leftarrow \text{MakeNode}([\text{InitialState}[\text{problem}])
\]
\[
\text{for } t \leftarrow 1 \text{ to } \infty \text{ do}
\]
\[
T \leftarrow \text{schedule}[t]
\]
\[
\text{if } T = 0 \text{ then}
\]
\[
\text{return current}
\]
\[
\text{end if}
\]
\[
next \leftarrow \text{a randomly selected successor of current}
\]
\[
\Delta E \leftarrow \text{Value}[next] - \text{Value}[current]
\]
\[
\text{if } \Delta E > 0 \text{ then}
\]
\[
current \leftarrow next
\]
\[
\text{else}
\]
\[
current \leftarrow next \text{ with probability } e^{\Delta E / T}
\]
\[
\text{end if}
\]
\[
\text{end for}
\]

An EA can be configured to be functionally equivalent to SA as follows:

<table>
<thead>
<tr>
<th>Population size</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offspring size</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Initialization</th>
<th>the single pop member is uniform randomly selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent selection</td>
<td>n/a (there is only one potential parent)</td>
</tr>
<tr>
<td>Recombination rate</td>
<td>0%</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mutation</th>
<th>offspring = random successor of the parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survival selection type</td>
<td>(\mu + \lambda)</td>
</tr>
<tr>
<td>Survival selection</td>
<td>IF F[offspring] - F[parent] &gt; 0 THEN offspring survives ELSE offspring survives with probability e^{F[offspring] - F[parent] / T}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Termination condition</th>
<th>\text{T}==0</th>
</tr>
</thead>
</table>