Introduction to Evolutionary Computation and Evolutionary Computation Module
Extra Tutorial
Exercises 9 and 10

V. Landassuri-Moreno
v.landassuri-moreno@cs.bham.ac.uk

School of Computer Science
University of Birmingham

December 11, 2009

Outline

Q1. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique

- Inspired by social behavior of bird flocking or fish schooling
- Similar to EA where a population is randomly generated and updated every generation looking for the optimal
- However there is no Recombination or Mutation operators
- Here the solutions are called “particles”, and they fly through the search space by following the best particles (in a given neighbourhood)

Q2. Differential Evolution

DE is similar to PSO, where it is a population base algorithm which is initialized at random and updated at each generation to look for the optima.

- Distance and direction information from the current population is used to guide the search process
- Here is also applied the selection, crossover and mutation operators but with a few differences
- Mutation: At each generation, new vectors (trial vectors) are generated by the combination of vectors randomly chosen from the current population and obtaining the difference of them
- Crossover: The out coming vectors (trial vectors) are then mixed with a predetermined target vector
- Selection: The trial vector is accepted iff (if and only if) it improves the solution of the objective function (greedy algorithm)
- DE is capable of handling non-differentiable, nonlinear and multimodal objective functions

Q3. *Local search* and *greedy criterion*

- Local Search is to have a method that let us optimize an objective function, thus is generated new solutions to see if they improve the actual one (best found so far) through a search space
- A direct search algorithm must have a strategy to generate variations of the parameter vector, thus decide if the new solution must be accepted or not [1]
- Then, we can say that the concept of *Local search* and *Direct search* are similar
- Usually a generic local search accept the new solution iff it improves the solution (greedy criterion)
- We can use direct search when the function to optimize is non-differentiable and nonlinear
- But with the greedy criterion it is possible that the algorithm falls in a local optima. There are other techniques that improve the greedy criterion as Simulated Annealing

[1] Definition form paper given en lectures: Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces

Q4. Simulated Annealing (SA)

- Comes from annealing in metallurgy, where it is a technique involving heating and controlled cooling of materials to increment the size of their crystal and reduce their defects
- The SA algorithm replaces the current solution by generating random number is accordance with a value called “Temperature” (T), which is gradually reduced during the process (cooling)
- At the beginning the SA algorithm could be seen with a random behavior, but as the iterations advance it starts to move downhill as the Temperature goes to zero

Q5. Lamarkian evolution and Baldwin effect

Lamarckian evolution

- The information of parents gained during their life are passed to the offspring, e.g. evolving ANNs, the weights of parents are used to generate offspring
- Note that here we are focus in the information learn by parents, which is the information passed to offspring. That is different to the information carried in the genes of the parents, which could represent layers, nodes and connections in the case of ANNs

Baldwin effect

- The information already gained is not passed to further generations, e.g. evolving ANNs, at each new generations is generated new random weights.

For some functions it is possible to converge to a local optima with the lamarckian method whereas with the bladwin method is possible to converge to global optima. At the end that is problem dependent and you need to choose the best that suits your requirements.

Q6. EAs' limitations

Here are presented some limitations from EAs.
It could be a good idea to have them in mind when you design an EA.

- It is required the selection of different parameters: population size, crossover and mutation probabilities, selection pressure, among others
- These parameters are not arbitrary and they are problem dependent
- EAs require more computation than other problems
  - e.g. problems that use the information of the gradient or other mathematical solutions
- It is not warranty that you find the optimal, even though the best solution found could be close to the optima
- Every time you run the algorithm, it used to give different values. If the objective function is easy, probably you will have the same values or close to previous runs.

Q6.

- There is not a general algorithm that is superior for all objective functions / solution spaces (related with the NFL theorem). Thus you need to choose one algorithm to solve a particular task (there is an advantage if you have previous domain knowledge of the problem).
- EAs could be trapped in a local minima. If the algorithm finish and give you the final result, can you know if the algorithm fell in a local minima?
- If the elitism is not used, we can lose the best individual(s) found
- EAs used to be computationally expensive in some cases
- Some times it can be difficult to design good fitness functions

Some answers comes from your answers (to have many different point of views - do you agree with all of them?)
After all this negative aspects, could you say why EAs are very useful techniques?

Q1. Markov Chain and Simple random walk

Markov Chain

- A common definition: given the current state, future states in the system will be independent of past states
- From another point of view, there is no history of the system, because all the information needed to generate next steps are in the present state, then it can be seen as a probabilistic behavior for future states

Simple random walk

- Imaging that the state space is a matrix, so in a random walk to move from the current state (current position) to another one, means to move to any of the neighbors of the current position, regardless what happen before

Q2. Monte Carlo method and pseudo code

- Based in repeated random sampling to calculate the results
- Used when is not possible to apply a deterministic algorithm
- There is not a single Monte Carlo Method, but a general framework could be given:
  - Define inputs
  - Generate the random inputs
  - Calculate a deterministic function
  - Aggregate the individual computation to the final result

Q3. EDAs (Estimation of Distribution Algorithms)

- It has a population of candidate solutions.
- EDA replace standard Crossover and Mutation Operators by building probabilistic model of selected solutions.
- Then it sample the built model to generate new solutions, e.g. for a bit string of 5 bits, is created a vector of probabilities $p(p_1, p_2, p_3, p_4, p_5)$. Then with the probability vector is possible to create an arbitrary number of individuals.
- They are superior to other EA because EDAs can adapt to the problem and deal effectively with both linkage learning and mixing.

Q4. How to generate a new individual given the empirical probability distribution

Given the empirical probability distribution $p(p_1, p_2, p_3) = p(5/12, 3/12, 4/12)$

- You can generate a random number x_1
- If $x_1 \leq p_1$ then you choose "0" else "1"
- That is repeat for each bit, for the second one, you generate another random number x_2
- If $x_2 \leq p_2$ then you choose "0" else "1"
- And so on for the entire individual.

Summary

- Through the tutorials you have seen different EAs in different languages, something useful to take them as a base (understand and improve them) or for a further development (plug new modules / functions)
- Many different concepts where reviewed during the term which where intended to improve your skills in the field
- Some questions came from different paper, important to improve your skills reading publications and getting used with them
- With all the information you have learn, now you have:
  - A wide overview of EAs
  - A particular knowled of different approaches / algorithms and state of the art information
  - More tools to tackle problems of the real life
  - The knowledge and diverse techniques to start a new research in EAs