**Assignment 05 from Lecture 06 (Sep 10, 2022)**

Submission Link: <https://froms/gle/4icummZvjnwxnKte8>

Deadline: September 16, 2022 9:00PM

1. Report Submission (File Name: Rool Number\_)

Q1. Write Statements and Short Notes from the given Two Videos regarding Swam Intelligence

**Answer Hints:**

Video 01:

|  |  |  |
| --- | --- | --- |
| SI. | Statement | Time Location in Video |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Video 02:

|  |  |  |
| --- | --- | --- |
| SI. | Statement | Time Location in Video |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Short Note:

Q2. Why we prefer non-deterministic features over deterministic features in the population based algorithms?

How nondeterministic features are assured in PSO and GA?

**Answer:-**

**Population based algorithms:**

The population is usually defined as a two dimensional array of – size population, size x, chromosome size.

Population Initialization

There are two primary methods to initialize a population in a GA. They are −

Random Initialization − Populate the initial population with completely random solutions.

Heuristic initialization − Populate the initial population using a known heuristic for the problem.

Population is a subset of solutions in the current generation. It can also be defined as a set of chromosomes. There are several things to be kept in mind when dealing with GA population −

The diversity of the population should be maintained otherwise it might lead to premature convergence.

The population size should not be kept very large as it can cause a GA to slow down, while a smaller population might not be enough for a good mating pool. Therefore, an optimal population size needs to be decided by trial and error.

**Deterministic features:**

In a deterministic algorithm, for a given particular input, the computer will always produce the same output going through the same states but in the case of a non-deterministic algorithm, for the same input, the compiler may produce different output in different runs. In fact, deterministic algorithms can solve the problem in polynomial time and can determine what the next step is, that may not be the best probable solution.

**Nondeterministic features:**

The non-deterministic algorithms can show different behaviors for the same input on different execution and there is a degree of randomness to it. A nondeterministic algorithm is an algorithm that can exhibit different behaviors on different runs, as opposed to a deterministic algorithm. A concurrent algorithm can perform differently on different runs due to a race condition.

A probabilistic algorithm's behaviors depend on a random number generator. An algorithm that solves a problem in nondeterministic polynomial time can run in polynomial time or exponential time depending on the choices it makes during execution. Non-deterministic algorithms are useful for finding approximate solutions when an exact solution is far too difficult or expensive to derive using a deterministic algorithm. While a deterministic algorithm will only produce a single output of the same input even on several different runs, a non-deterministic algorithm would travel through various routes, allowing it to reach multiple different outcomes.

Non-deterministic models are primarily used when the problem that the algorithm seeks to solve inherently allows multiple outcomes or when there is a single outcome that can be found by going down multiple paths, and each of the paths that could be followed is equally preferable. Essentially, every outcome that the non-deterministic algorithm could produce is valid. These outcomes are valid irrespective of the choices and decisions that the algorithm makes while running and making an attempt to solve the problem. Algorithms for which the result of every algorithm is uniquely defined are known as deterministic algorithms. In the theoretical framework, it is possible to get rid of this restriction on the outcome of every operation. It is possible to allow algorithms to contain operations, the outcomes of which are not uniquely defined but are instead limited to specified sets of possibilities. The machine that executes every operation will be allowed to select any of these outcomes, subject to a determination condition that will be defined at a later point. This is what leads to the concept of non-deterministic algorithms.

So, in summery we can say that – With deterministic models arriving at best result would have been impossible because, in a deterministic algorithm, for a given particular input, the computer will always produce the same output. To maintain the diversity of the population and to prevent premature convergence, to maintain a proper population enough for a good mating pool i.e.- optimal population size to determine the best probable solution by trial and error method with multiple stages of random population selections, crossovers and mutations we need non-deterministic algorithms.

**How nondeterministic features are assured in PSO and GA:**

**Assuring non-deterministic feature in PSO:** In PSO algorithm the main approach to maintain non-deterministic feature is to find different velocity on each iteration. In velocity update equation we have user defined constant which we cannot change but we also use a random number which we can control. So by using different random numbers on each iteration we can find different result thus establishing the non-deterministic property in PSO.

**Assuring non-deterministic feature in GA:** GA is a population based algorithm, to maintain the non-deterministic property we need to start from a population instead of a single point in the potential solution space of a specific problem and allow the population to evolve from generation to generation by generic operators such as selection, crossover, and mutation until the stopping criteria are satisfied.

Q3. List of Three recent (not before 2017) selected Journal Articles on interesting/important problem solving with PSO.

**Answer:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SI** | **Year of Publication** | **Article Title** | **Name of the Journal** | **Volume, Issue and Page** |
| 1 | 2020 | Service Composition in IoT using Genetic algorithm and Particle swarm optimization | Open Computer Science | Volume10, Issue 1 and Page 56-64 |
| 2 | 2021 | An unknown fault identification method based on PSO-SVDD in the IoT environment | Alexandria Engineering Journal | Volume 60, Issue 4 and Page 4047– 4056 |
| 3 | 2021 | Research on Intrusion Detection Based on Particle Swarm Optimization in IoT | IEEE Access | Volume 9, Page 38254-38268 |

1. **Uploaded Zip file having the Three Selected Articles (File Name : MCE 079 05536 Shyed Shahriar Housaini Assignment 05)**