Lab Topic 5

Copyright: Soumya D. Mohanty

PSO codes

Clone the repository: **GitHub →SDMBIGDAT19** (Store it outside SandBox)

The codes and slides supplement the textbook

Lectures delivered at the BigDat19 5th International Winter school on Big Data, Cambridge University, UK (Jan, 2019)

http://bigdat2019.irdta.eu/

We will look at the following codes in **SDMBIGDAT19/CODES**:

▶ **r2ss.m:** Helper function; no need to look inside

▶ **r2sv.m:** Helper function; no need to look inside

▶ **s2rs.m:** Helper function; no need to look inside

▶ **s2rv.m:** Helper function; no need to look inside

▶ **crcbchkstdsrchrng.m:** Helper function; no need to look inside

▶ **crcbpso.m: Main PSO code that can be applied to any fitness function**

▶ **crcbpsotestfunc.m:** A benchmark fitness function; Also an example for how to code fitness functions to work with crcbpso.m

▶ **crcbqcfitfunc.m:** The fitness function for quadratic chirp GLRT (in WGN)

▶ **crcbqcpso.m**: Applies PSO to the quadratic chirp fitness function

▶ **test_crcbpso.m:** Test function for crcbpso.m

▶ **test_crcbqcpso.m:** Test function for crcbqcpso.m

Exercise #1 Part 1

- Read the short user manual **CODES/CodeDoc.pdf**

- The main usage instructions are in the "help" for each function

- The **test_<funcName>.m** scripts show examples of usage for some of the functions

    - The **test_crcbpso.m** script shows how **crcbpso.m** is applied to a benchmark fitness function (defined in **crcbpsotestfunc.m**)

Exercise #1 Part 2

- Understand the concept of structures in Matlab
  - Matlab structures work in the same way as structures in C
  - X = struct('a', 5.0, 'b', 6.0);
  - disp(X.a) will show 5.0
  - disp(X.b) will show 6.0
- Structures offer a convenient way to move a large number of arguments into and out of a function
- Structures also help make your codes future-proof: New versions of codes can use new input arguments while old versions will ignore them

Exercise #1 Part 3

▶ Understand the concept of **function handle**

▶ A function handle is a variable that can be used to call a function

▶ $z = 5.0$

▶ $F = \underbrace{@(x, y)}_{\substack{\textit{What are the input} \\ \textit{variables that will be} \\ \textit{sent to F?}}} foo(x, z, y)$

▶ F is a handle to function foo

▶ $F(2.0, 3.0)$ is the same as $foo(2.0,\ 5.0,\ 3.0)$

▶ The **CRCBPSO** code accepts as input the handle to the fitness function to be optimized

▶ This allows the same PSO code to be run on any fitness function

▶ Type "function handle" in Matlab's "Search documentation" bar and read more about this feature