Matlab Code Explanation

Matpower

- simulating and optimizing steady-state operations of electric power systems
- **Offers**
    - Power Flow (PF): Solving AC and DC power flow problems to determine voltage magnitudes and angles across the network.
    - Continuation Power Flow (CPF): Analyzing voltage stability and system loading limits.
    - Optimal Power Flow (OPF): Determining the most economical operating conditions while satisfying system constraints.
    - Unit Commitment (UC): Scheduling generation units to meet demand over a time horizon at minimal cost.
    - Stochastic and Security-Constrained OPF/UC: Incorporating uncertainties and contingencies into planning and operation.
- ## Structure
    - MP-Core: The object-oriented core providing a flexible framework for modeling and simulation.
    - MIPS (MATPOWER Interior Point Solver): A solver for nonlinear programming problems, particularly OPF.
    - MP-Opt-Model: A tool for constructing and solving mathematical optimization problems.
    - MOST (MATPOWER Optimal Scheduling Tool): A framework for generalized steady-state electric power scheduling problems.

CODE

Some defs.

- Bus Data

```
% bus_i  type  Pd   Qd   Gs  Bs  area  Vm   Va  baseKV  zone  Vmax  Vmin
```

Field	Description
`bus_i`	Bus number (unique integer ID)
`type`	Bus type: 1 = PQ (load), 2 = PV (generator), 3 = Slack (reference)

Field	Description
`Pd`	Real power demand at bus (MW)
`Qd`	Reactive power demand at bus (MVAr)
`Gs`	Shunt conductance (MW at V = 1.0 p.u.)
`Bs`	Shunt susceptance (MVAr at V = 1.0 p.u.)
`area`	Area number (for area-based control; rarely used)
`Vm`	Voltage magnitude (p.u.)
`Va`	Voltage angle (degrees)
`baseKV`	Base voltage level of the bus (kV)
`zone`	Loss zone (used in some optimization cases)
`Vmax`	Maximum voltage magnitude (p.u.)
`Vmin`	Minimum voltage magnitude (p.u.)

- Generator Data

```
%bus Pg Qg Qmax Qmin Vg mBase status Pmax Pmin
Pc1,Pc2,Qc1min,Qc1max,Qc2min,Qc2max,ramp_agc,ramp_10,ramp_30,ramp_q,apf
```

Field	Description
`bus`	Bus number at which the generator is connected
`Pg`	Real power output (MW)
`Qg`	Reactive power output (MVAr)
`Qmax`	Maximum reactive power output (MVAr)
`Qmin`	Minimum reactive power output (MVAr)
`Vg`	Voltage magnitude setpoint (p.u.)
`mBase`	Machine base MVA (used for per-unit conversion; often same as system base)
`status`	Generator status (1 = in service, 0 = out of service)
`Pmax`	Maximum real power output (MW)
`Pmin`	Minimum real power output (MW)
`Pc1-Qc2max`	Piecewise linear cost and reactive power curve (optional; for OPF)
`ramp_*`	Ramping limits (AGC, 10-min, 30-min, Q, etc.)
`apf`	Area participation factor (for AGC control)

- Branch Data

```
% fbus tbus r x b rateA rateB rateC ratio angle status angmin angmax
```

Field	Description
`fbus`	"From" bus number
`tbus`	"To" bus number
`r`	Resistance (p.u. on system base)
`x`	Reactance (p.u.)
`b`	Total line charging susceptance (p.u.)
`rateA`	MVA rating A (continuous thermal limit)
`rateB`	MVA rating B (emergency/short-term limit)
`rateC`	MVA rating C (worst-case limit)
`ratio`	Transformer off-nominal turns ratio (if not a transformer, use 0 or 1)
`angle`	Transformer phase shift angle (degrees)
`status`	Branch status (1 = in service, 0 = out of service)
`angmin`	Minimum angle difference across branch (degrees)
`angmax`	Maximum angle difference across branch (degrees)

How it works/What it does.

```
function mpc = case33Loss202
```

- Matlab function defined called *"case33Loss202"*, outputs a specific mpc struct
- represents a standard 33-bus test system used in loss minimization studies in distribution systems.

```
mpc.version = '2';
mpc.baseMVA = 100;
```

- mpc version 2 mentioned but newer versions exsist?
- All power values are normalised to 100MVA
  After this bus data, generator data and branch data are used [Here]

Output

```
case33Loss202


ans =
```

```
[struct](matlab:helpPopup\('struct'\)) with fields:

version: '2'
baseMVA: 100
bus: [33×13 double]
gen: [33×21 double]
branch: [32×13 double]
gencost: [66×7 double]
```

- All the input given is being read correctly by matpower
- For more details look into run_case33loss202, Contains a basic test case

MVO Code:

- For details on MVO look into Optimisation.
- Most of this code is well documented so this can be skipped and directly the .m file can be looked into

  ```
  sd = 2:33;
  ```

  Mentions the available number of buses, bus 1 is ignored as it is usually a slack bus

  ```
  dim1 = 3;
  ```

  tells that 3 devices need to be placed

  ```
  Saizi = 0.15:0.15:4.05;
  ```

  Capacitor Size changes.
  The above can be classified as the search space.

  ```
  Max_iteration = 100;
  SearchAgents_no = 10;
  WEP_Max = 1;
  WEP_Min = 0.2;
  ```

  MVO Parameters
- After this boundaries are defined for
  - Bus Combination
  - Capacitor banks

```
Best_universe = zeros(1,dim1);

Best_universe1 = zeros(1,dim1);

Best_universe_Inflation_rate = inf;
```

Contains the solutions both position and actual value

- After this randomly bus positions are assigned and snapped to nearest bus number as defined.
- Similarly capacitors are randomly initialised and snapped to nearest size as defined.
- After this MVO optimisation takes place WEP, TDR are updated with `p = 6`
- Fitness is evaluated using `case33Loss202`.
- Voltage constraints are checked.
- Fitness function evaluated
    - Minimizes total real power losses across branches.
    - If voltage constraints are not met, penalty is applied to fitness:
- Universe sorted, inflation rates normalised.
- Position update (MVO).
- Final results displayed
- Output came first time but after that didnt come

References

[MATPOWER Documentation — MATPOWER Documentation 8.0 documentation]