Particle Swarm Optimzation: A Brief Guide

anaSilva2018

14th October 2020

1 What is particle swarm optimization?

Based on swarm intelligence, the particle swarm optimization(PSO) is a meta-heuristic optimization algorithm non-evolutionary.

1.1 Main parameters

1. **Chromosome**: set of real variables/characteristics. In this example, the variables are the pumped volume for which time period i (m = 6).

2. **Particles** : Which particle j is defined by m variables (1 chromosome). In this example, the total number of particles is 20 (n = 20).

3. **Generations**: Number of iterations. In which one is possible to find a new particle with the optimal cost. In this case, iterations equal to 1000 (p = 1000).

4. **Fitness**: In this case corresponds to the minimization of power generation, which includes penalties for the violations of constraints (PEN), water storage rewards on last time period (VAL), and direct costs of thermal power generation (CP). *Constraints: Limits of thermal power production ($0 - 80MW$) and of pumped volume ($0 - 80000m^3$).*

$$MinF(C_{total}) = \sum CP(P_{thermal_i}) + \sum PEN(Const_j) - \sum VAL(P_{stored_k})$$

$$CP(P_{thermal_i}) = 2000 + 100P_{thermal_i} + 1.5P^2_{thermal_i}$$

$$PEN(Const_j) = 1000(P_j - Pref_j)^2$$

$$VAL(P_{stored_k}) = MarginalCost.P_{stored_k}$$

1.2 Steps

1. **Initial Swarm** (mPopula): Create two matrices: Initial Swarm (mPopula) and Predecessor Swarm(pop_ant). Predecessor Swarm is created after mutate the Initial Swarm. Which matrix is composed by 6 characteristics and 20 particles.

2. **Initial Evaluation**(_swarm_best): Compare the fitness functions of both swarms, in order to find the best particles(global and local).

3. **Move, Evaluate and Update**(_move_updatebest):

   Move: Update the velocity and position of particles.

   Evaluate: Evaluate the fitness function of swarm.

   Update: Update the global and local best.

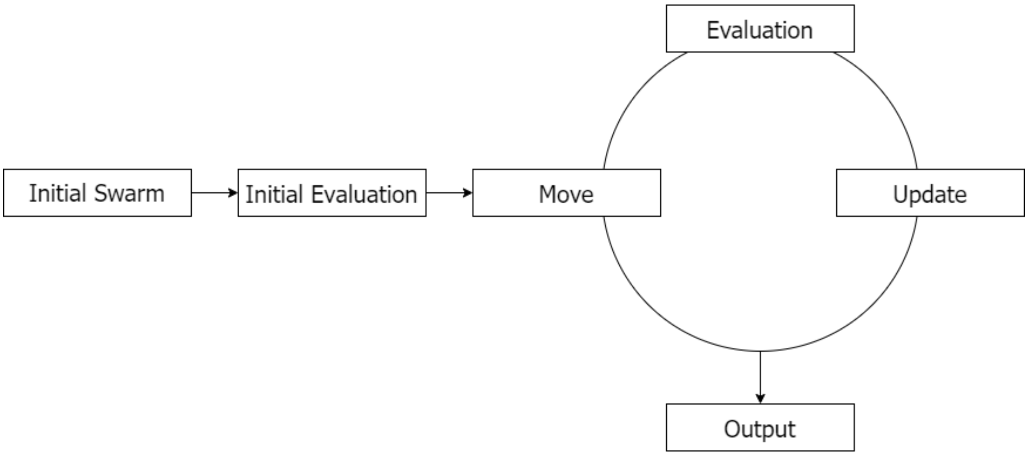

Figure 1: Particle Swarm Optimization flowchart.

2 Classes

_DataPop(sigma, pop, germax, perload, pervolin, perafl)

Definition: Class of population data.

sigma: fixed mutation tax.

pop: number of particles.

germax: maximum number of generations.

perload: per-unit value of load power.

pervolin: per-unit value of initial water volume, in the first period.

perafl: per-unit value of water affluence.

_DataHydTh(vdispmax, vturbmax, vinicial, penaliz, ptermax)

Definition: Class of Hydro-Thermal system.

vdispmax: maximum volume of water available (m^3).

vturbmax: maximum pumped volume (m^3).

vinicial: absolute value of initial water volume, in the first period (m^3).

penaliz: penalty value (€).

ptermax: maximum value of thermal power generation (MW).

_DataSwarm(_wi, _wm, _wc)

Definition: Class of swarm data.

_wi: inertia weight. Larger wi results in greater global search ability, and smaller wi results in greater local search ability.In this case, wi = 0.5.

_wm: memory weight. Define the particle's atraction for the local best(bi).In this case, wm = 2.

_wc: cooperation weight. Define the swarm's atraction for the global best(bg). In this case, wc = 2.

$$V1_{i,j} = \frac{1}{Ger+2}.\_wi.[mPopula(i,j) - pop\_ant(i,j)]$$

$$V1_{i,j} = V1_{i,j} + Rand.\_wm.[popbest(i,j) - mPopula(i,j)]$$

$$V1_{i,j} = V1_{i,j} + Rand.\_wc.[popbest(i,nbglobal) - mPopula(i,j)]$$

$$X1_{i,j} = K.V1_{i,j} + X0_{i,j}$$

$$K = \frac{2}{|2 - W - \sqrt{W^2 - 4W}|}, W = wm + wc$$

$$i : timeperiod; j : particle$$

_BestLocal(popbest, cmgbest, phidrbest, ptermbest, vturbbest, vdispbest, vsobrbest)

Definition: Class with best particles data.

popbest: matrix of population for best particles. ($p.u.$).

cmgbest: matrix of marginal cost for the best particles (€/MW).

phidrbest: matrix of pumped water power/hydro power production for the best particles(MW).

ptermbest: matrix of thermal power production for the best particles (MW).

vturbbest: matrix of pumped water volume for the best particles (m^3).

vdispbest: matrix of volume of water available for the best particles (m^3).

vsobrbest: matrix of remaining water volume for the best particles (m^3).

_BestGlobal(nbglobal, ncmgbglobal, ncostbglobal)

Definition: Class with global best data.

nbglobal: position of global best.

ncmgbglobal: marginal cost of global best (€/MW).

ncostbglobal: total cost of global best (€).

_PopVol(vherd, vdisp, vturb, vsobr, vdesc)

Definition: Class with volume data.

vherd: matrix of inherited water volume (m^3).

vdisp: matrix of volume of water available (m^3).

vturb: matrix of pumped water volume (m^3).

vsobr: matrix of remaining water volume (m^3).

vdesc: matrix of excessive water volume (m^3).

_PopPot(psobr, phidr, pterm)

Definition: Class with Hydro-Thermal data.

psobr: matrix of remaining water power (MW).

phidr: matrix of pumped water power/hydro power production (MW).

pterm: matrix of thermal power production (MW).

_Ccost(mcmg, mpen, mval, mcost)

Definition:Class with the cost data.

mcmg: matrix of marginal cost (€/MW).

mpen: matrix of penalties (€).

mval: matrix of remaining volume (m^3).

mcost: matrix of total cost (€).

_cGerb(gbest, gcost, gcmg, gphidr, gpterm, gvturb, gvsobr)

Definition:Class with generational evolution of the global best.

gbest: matrix of positions for the global best. ($p.u.$).

gcost: matrix of total cost for the global best. (€).

gcmg: matrix of marginal cost for the global best. (€/MW).

gphidr:matrix of pumped water power/hydro power production for the global best (MW).

gpterm: matrix of thermal power production for the global best (MW).

gvturb: matrix of pumped water volume for the global best (m^3).

gvsobr: matrix of remaining water volume for the global best (m^3).

3 Functions

_pop_mutate(pop, nper, mpop, sigma)

return: mmutate

Definition: Mutation.

pop: number of particles.

nper: number of time periods.

mpop: matrix of initial swarm ($p.u.$).

sigma: fixed mutation tax.

mmutate: matrix of predecessor swarm($p.u.$).

_pop_evaluate(nper, cinic, chydth, nvolinic, mafl, mutate, mload, valdef)

return: _PopVol,_PopPot, _Ccost

Definition: Evaluation.

nper: number of time periods.

cinic: Class with population data.

chydth: Class with hydro-thermal system.

nvolinic: initial volume (m^3).

mafl: matrix of water affluence (m^3).

mutate: matrix of swarm ($p.u.$).

mload: matrix of load power (MW).

valdef: marginal cost (€/MW).

_PopVol: Class with volume data.

_PopPot: Class with Hydro-Thermal data.

_Ccost: Class with the cost data.

_swarm_best(mpop_init, mpop_ant, nper, pop, cpinit, cvinit, ccinit, cpant, cvant, ccant)

return: _BestLocal, _BestGlobal

Definition: Initial Evaluation

mpop_init: matrix of initial swarm ($p.u.$).

mpop_ant: matrix of predecessor swarm ($p.u.$).

nper: number of time periods.

pop: number of particles.

cpinit: Class with Hydro-Thermal data of initial swarm.

cvinit: Class with volume data of initial swarm.

ccinit: Class with cost data of initial swarm.

cpant: Class with Hydro-Thermal data of predecessor swarm.

cvant: Class with volume data of predecessor swarm.

ccant: Class with cost data of predecessor swarm.

_BestLocal: Class with best particles data.

_BestGlobal: Class with global best data.

_move_updatebest(cbi, cbg, mpop, pop, nper, ngermax, cSwarm, mpop_ant, cpar, chydth, nvolinic, mafl, mload, valdef)

return: _cGerb

Definition: Move, Evaluate and Update.

cbi: Class with best particles data.

cbg: Class with global best data.

mpop: matrix of initial swarm ($p.u.$).

pop: number of particles.

nper: number of time periods.

ngermax: maximum number of generations.

cSwarm: Class of swarm data.

mpop_ant: matrix of predecessor swarm ($p.u.$).

cpar: Class of population data.

chydth: Class of Hydro-Thermal system.

nvolinic: initial volume (m^3).

mafl: matrix of water affluence (m^3).

mload: matrix of load power (MW).

valdef: marginal cost (€/MW).

_cGerb: Class with generational evolution of the global best.

4 Other variables

mdata_afl: matrix of water affluence (m^3).

mdata_load: matrix of load power (MW).

nPeriod: number of time periods.