Evolutionary Programming: A Brief Guide

anaSilva2018

13th October 2020

1 What is evolutionary programming?

Based on evolutionary principles, i.e. natural selection, the evolutionary programming (EP) is a meta-heuristic optimization algorithm.

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. **Individual** : Which individual j is defined by m variables (1 chromosome). In this example, the total number of individuals is 40 (n = 40), 20 parents and 20 successors.
3. **Generations**: Number of iterations. In which one is possible to find a new individual 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. **Initialisation** (mPopula): matrix composed by 6 characteristics and 20 parents.
2. **Duplication**(_pop_duplicate): copy the characteristics of parents to the successors, which means the creation of the successors. Matrix composed by 6 characteristics and 40 individuals.
3. **Mutation**(_pop_mutate): apply a mutation tax (*sigma*) to the successors, in order to create new characteristics.
4. **Evaluation**(_pop_evaluate): evaluate all the penalties, rewards and thermal power production costs from all individuals.
5. **Selection**(_cSelect): obtainment of the best individuals, which are the ones without any violations of constraints and with minimum total cost. They will pass its characteristics to the parents of the next generation (Elitist Selection).

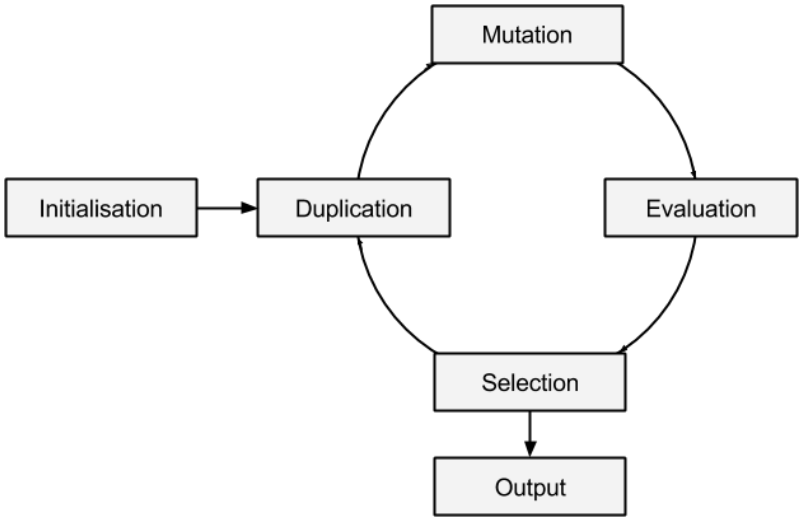

Figure 1: Evolutionary programming flowchart.

2 Classes

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

Definition: Class of population data.

sigma: fixed mutation tax.

pop: matrix of initial population ($p.u.$).

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).

_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 (€).

_PopSelect(mcaux, mvaux, mcostm, mvsel, newpop, best)

Definition: Class of best individuals data.

mcaux: matrix of total cost for the best individuals (€).

mvaux: matrix of pumped volume for the best individuals ($p.u.$).

mcostm: matrix of the new total cost (€).

mvsel: matrix of the new pumped volume ($p.u.$).

newpop: matrix of the new population of parents ($p.u.$).

best: last best individual found.

3 Functions

_pop_duplicate(mpop, nper, pop)

return: mduplicate

Definition: Duplication.

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

nper: number of time periods.

pop: number of parents.

mduplicate: matrix of duplicate population($p.u.$).

_pop_mutate(pop, nper, mduplicate, sigma)

return: mmutate

Definition: Mutation.

pop: number of parents.

nper: number of time periods.

mduplicate: matrix of duplicate population ($p.u.$)

sigma: fixed mutation tax.

mmutate: matrix of mutated population($p.u.$).

_pop_evaluate(nper, cinic, chydth, nvolinic, mafl, mmutate, 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.

mafl: matrix of water affluence (m^3)

mmutate: matrix of mutated population ($p.u.$)

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.

_cSelect(pop, nper, mmutate, mcost)

return: _PopSelect

Definition: Selection.

pop: number of parents.

nper: number of time periods.

mmutate: matrix of mutated population ($p.u.$)

mcost: matrix of total cost (€).

_PopSelect: Class of best individuals data.

4 Other variables

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

mdata_load: matrix of load power (MW).

nPeriod: number of time periods.