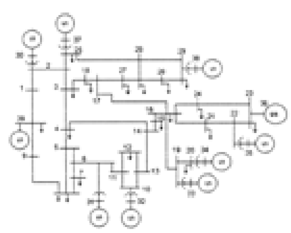
Coordinated Tuning of Power System Stabilizers using Parallel Genetic Algorithm

Prof. Carmen L. T. Borges
carmen@dee.ufrj.br

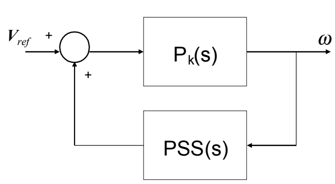
Federal University of Rio de Janeiro
P.O.Box 68516, 21945-970
Brazil

Problem Formulation (1)



- *Electrical Power System*
- *Determined number of operating conditions*

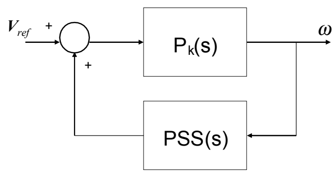
Problem Formulation (2)



$$PSS(s) = diag[PSS_i(s)]$$

$i = 1, 2, \ldots, p$
($p \rightarrow$ number of stabilizers)

Problem Formulation (3)



$\boldsymbol{P_{\mathrm{k}}(s)}$ k **= 1, 2, …,** $\boldsymbol{m}$

($\boldsymbol{m}$ **→ number of operating conditions)**

Otimization Problem

$$\text{Max} \quad F = \sum_{k=1}^{m} \left[\sum_{j=1}^{n+3\times p} (\zeta_j) \right]_k$$

Subject to:

- Max and min limits of PSS parameters K, α e ω
- Min value of damping ζ_{min}

Where:

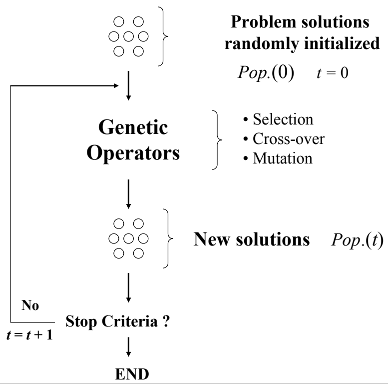
n – no. of linearized system states
F –fitness function value
ζ – damping value

Genetic Algorithm

- **Optimization algorithm**
- **Based on natural evolution of species**
- **Independent of objective function**
- **Evolutionary computation**
- **Genetic operators**

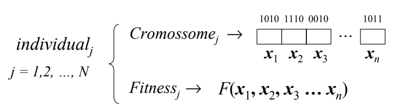
Coordinated Tuning of Power System Stabilizers using Parallel Genetic Algorithm

Genetic Algorithm



GA Characteristics

- Work with a lot of feasible solutions (individuals)

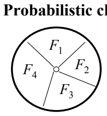


- Several codifications (binary,real).
- Probabilistic search rules (not deterministic)

Genetic Operators

Selection:

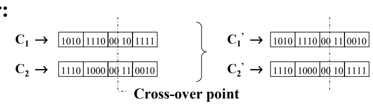
Probabilistic choice of the individuals with higher fitness



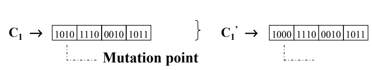
Example: $N = 4$ where: $F_4 > F_3 > F_1 > F_2$

New generation formed by the selected individuals

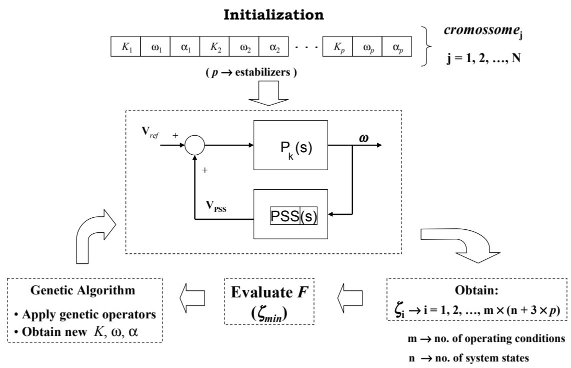
Cross-over:



Mutation:



GA applied to PSS Tuning



Undesirable Characteristic

GA → Elevated computation time

Solution:

Parallel Genetic Algorithm (PGA)

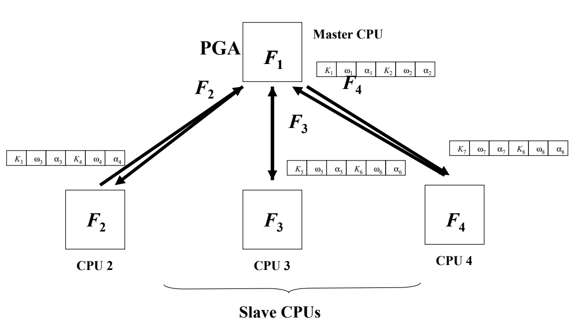
Strategies:

1. **Master-Slave**
   - **Individuals fitness evaluation in parallel**

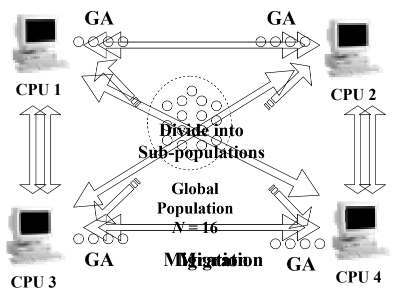
   (Reduce computation time / same solution)
2. **Multi-Population**
   - **Divide population into sub-populations**

   (Reduce computation time / better solution (higher fitness)

Master-Slave PGA



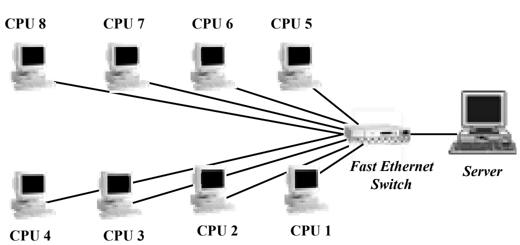
Multi-Population PGA



Multi-Population PGA Characteristics

- Migration Rate
  - number of individuals communicated between populations at each migration
- Migration Interval
  - number of generations between successive migrations
- Reception Strategy
  - rules for incorporating individuals migrated from other populations
- Connectivity
  - number of CPUs involved in the migration from each CPU

The Cluster Used

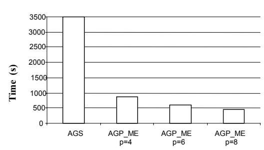


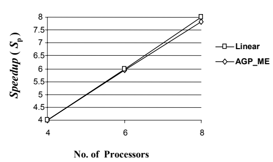
- 8 PCs PENTIUM III / 500 MHz / 128 MB RAM
- Windows NT 4.0
- MPI

Master-Slave PGA Results

Algorithm	P	$t_{med}(s)$	f_{max}	ζ_{min}	s_p	$e(\%)$
AGS	–	3494.825	589.98	0.1503	–	–
AGP_ME	4	876.897	589.98	0.1503	3.985	99.64
AGP_ME	6	589.318	589.98	0.1503	5.930	98.84
AGP_ME	8	446.298	589.98	0.1503	7.831	97.88

Speedup





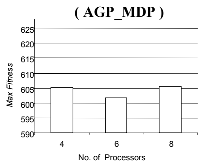
Multi-Population PGA Results

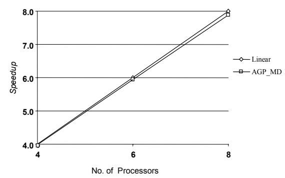
Algorithm	P	$t_{med}(s)$	f_{max}	ζ_{min}	s_p	$e(\%)$
AGP_MDP	4	879.935	605.18	0.1537	3.972	99.29
AGP_MDP	6	587.519	601.66	0.1566	5.948	99.14
AGP_MDP	8	443.914	605.51	0.1624	7.873	98.41

Migration Rate:
1 *individual*

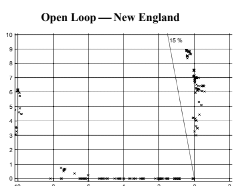
Full
Conectivity

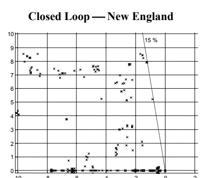
Max Fitness and Speedup





Poles Locus



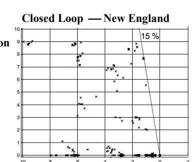


Master-Slave PGA

***P* = 8**

Multi-Population PGA

***P* = 8**



Conclusion

1 PGA can be fairly used for stabilizers tuning.

2 Almost linear speedup. Very high efficiency on the cluster.

3 Multi-Population PGA reduces computation time AND enhances the solution.