MEMETIC-BASED FAST AND ACCURATE DESIGN OPTIMIZATION FOR WIIMAX AND MMDS PLL

O. Garitselov[1], S. P. Mohanty[2], E. Kougianos[3], and O. Okobiah[4]

NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu) University of North Texas, Denton, TX 76203, USA.[1,2,3,4]

Email: omg0006@unt.edu[1], saraju.mohanty@unt.edu[2], eliask@unt.edu[3], and , oo0032@unt.edu[4]

Presented By
O. Okobiah
University of North Texas
Email: oo0032@unt.edu

Acknowledgment: This research is supported in part by NSF awards CNS-0854182 and DUE-0942629 and SRC award P10883.

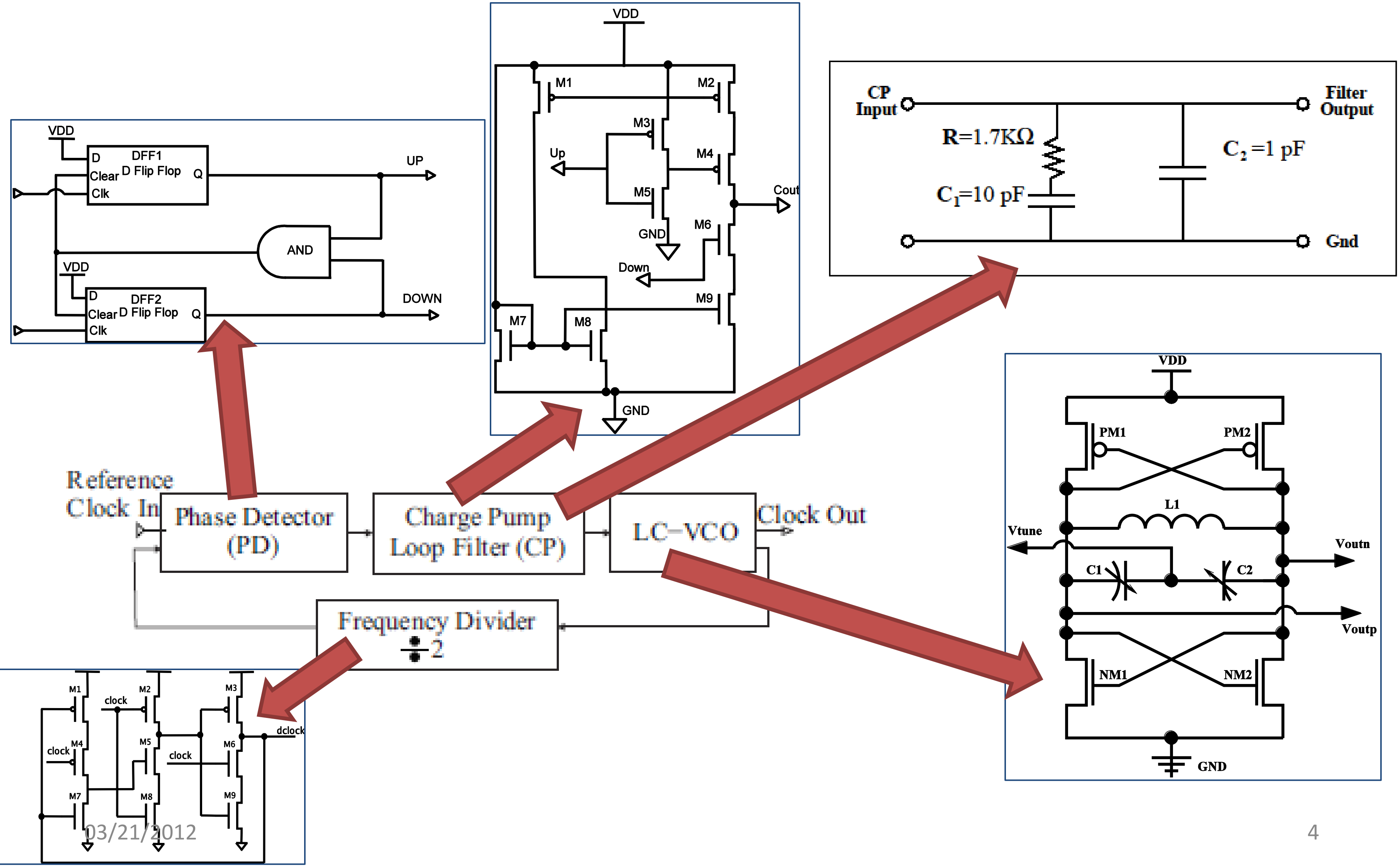
Outline of the talk

- Background and Motivation
- Issues in Nano-CMOS
- PLL Circuit
- Proposed Design Optimization Flow Methodology
- Memetic Optimization Algorithm and Optimization Results
- Conclusions and Future Research Directions

Introduction

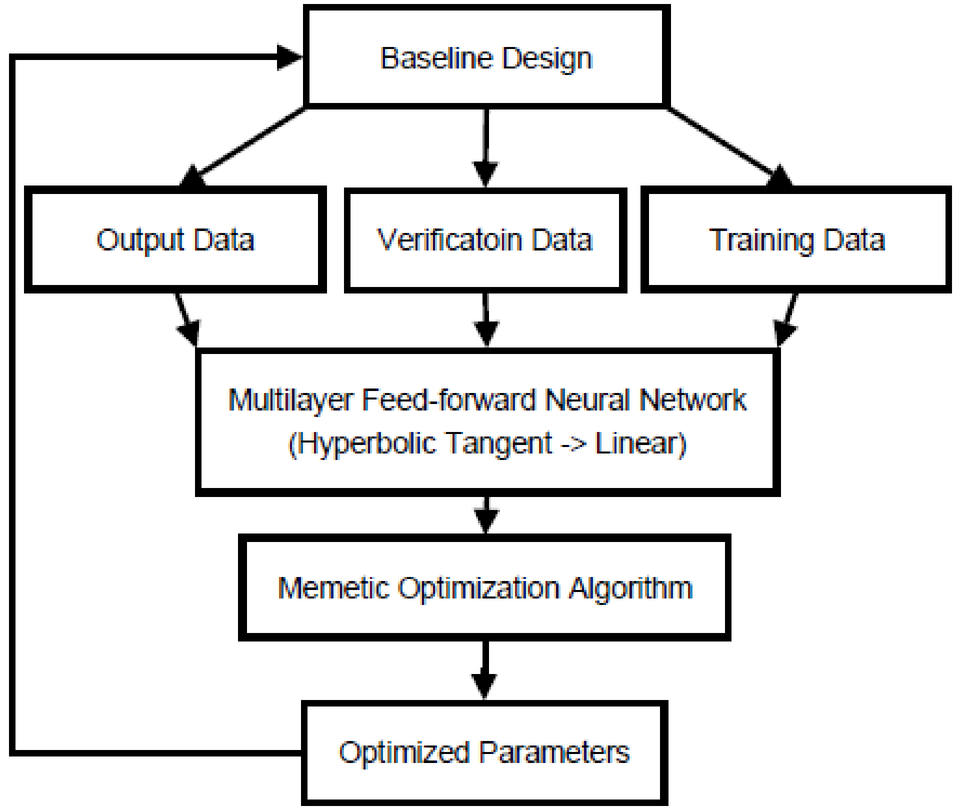
- Analog and AMS simulations involve more intense calculations, which makes simulation time consuming.
- Conducting optimization on physical layout requires multiple simulations and is very costly process in original design flow.
- Design flow using metamodels is introduced to reduce the design time.
- Metamodeling is mathematical formula that represents circuit behavior within a given range using sampling points.
- Successful sampling techniques are needed to reduce the amount of data samples.
- Design constraints placed on a design require accurate metamodels to increase yield of the process.

Phase Locked Loop 180 nm



Proposed Design Flow

- 100 training and 30 verification sample points are used for PLL.



Parameters Considered in PLL

Large range for 21 parameters for transistor sizing are considered.

Circuit	Parameter	Min (m)	Max (m)	Optimal Value (m)
Phase Detector	W_{ppd1}	400n	2μ	1.66μ
	W_{npd1}	400n	2μ	1.11μ
	W_{ppd2}	400n	2μ	784n
	W_{npd2}	400n	2μ	689n
	W_{ppd3}	400n	2μ	1.54μ
	W_{npd3}	400n	2μ	737n
Charge Pump	W_{nCP1}	400n	2μ	1.24μ
	W_{pCP1}	400n	2μ	1.35μ
	W_{nCP2}	1μ	4μ	1.35μ
	W_{pCP2}	1μ	4μ	2.88μ
LC-VCO	W_{nLC}	3μ	20μ	18.62μ
	W_{pLC}	6μ	40μ	37.48μ
Divider	W_{p1Div}	400n	2μ	1.65μ
	W_{p2Div}	400n	2μ	1.54μ
	W_{p3Div}	400n	2μ	1.38μ
	W_{p4Div}	400n	2μ	1.96μ
	W_{n1Div}	400n	2μ	1.09μ
	W_{n2Div}	400n	2μ	1.17μ
	W_{n3Div}	400n	2μ	1.29μ
	W_{n4Div}	400n	2μ	1.95μ
	W_{n5Div}	400n	2μ	536n

Metamodeling

- Metamodels are mathematical function(s) that are used to represent output of the circuit i.e. polynomial functions, Kriging interpolation, neural networks, and DOE predictive functions.
- Metamodels are created from the sampled data.
- The accuracy of metamodel is dependant on the amount of samples and its architecture.

The Metamodels

- Prediction equation:

$$(\hat{F}(x_n) \approx F(x_n)) = F(x_n) + \varepsilon$$

- Can be used in different tool like MATLAB.
- Metamodels are not as computationally expensive than conducting simulations.

Error Analysis

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y(x_k) - \hat{y}(x_k))^2},$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\left|y(x_k) - \hat{y}(x_k)\right| - RMSE)^2},$$

Where:

x_k is the set of parameters

$\hat{y}$ is the predicted equation

N is the number of sampling points

Error Analysis

Coefficient of determination R^2 predicts the probability of the future result to be predicted accurately by the metamodel. It ranges from 0 to 1, with 1 being the best result.

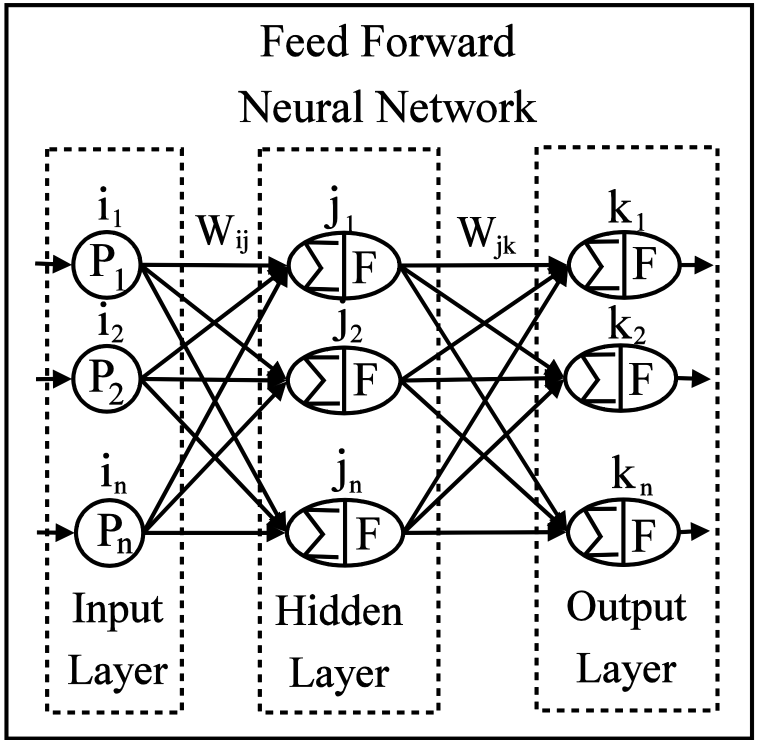
$$R^2 = \frac{SS_{err}}{SS_{tot}} = \frac{\sum (y_i - f_i)^2}{\sum (f_i - \bar{y})^2}$$

$R^2{}_{adj}$ accounts for the number of explanatory terms in a model.

$$R^2{}_{adj} = (1 - \frac{SS_{err}}{SS_{tot}} \frac{dt_t}{df_c})$$

Neural Network Models

- Feed-forward dual layer NNs (FFDL) are considered.
- FFDL network created for each FoM:
  - non-linear hidden layer functions are considered each varying hidden neurons 1-20:
    - A) $b_j\left(v_j\right) = \tanh(\lambda v_j)$



Memetic Algorithm

- R.Dawkin proposed the notion of meme in 1976.
- In 1989 P. Moscato has introduced first memetic algorithm.

Algorithm 4 Global Optimizer for Heuristic Memetic Algorithm.

```
Initialize initial population.
Initialize weights.
count = 0.
while count ¡ max_iterations do
  count=count+1.
  Evaluate all individual population.
  for each individual in the population do
    Select suitable memes.
    Process with Alg. 2 for local improvements and
    Replace selected meme with the improved solution.
    Receive information of the improved location for that meme.
    Adjust meme's weights.
  end for
  Calculate probability for random selection operations = prob.
  if prob is crossover then
    Swap selected meme's parameters.
  else if prob is mutation then
    Replace selected meme with random solution.
    Reset selected meme's weights.
  end if
end while
Return optimal value and location.
```

Memetic Algorithm

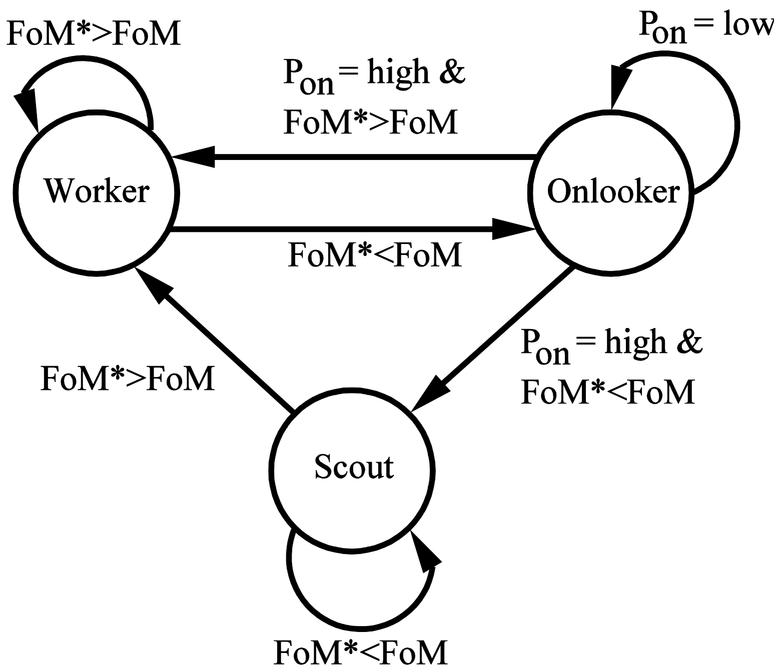
Algorithm 5 FoM function for MMDS application

```
Receive P coordinates.
Calculate frequency from neural network with P parameters = freq.
if freq is within specification then
    Calculate locking time from neural network with P parameters = l_time.
    if l_time ¡ specifications then
        Calculate power from neural network with P parameters = power.
        FoM = 1/(power*1e3).
    else
        FoM = 0.
    end if
end if
Return FoM
```

TABLE 7.12. Final Optimization Results for different FoMs for WiiMAN and MMDS PLLs.

	MMDS		WiiMAX	
	Memetic	ABC	Memetic	ABC
Power	0.51 mW	0.68 mW	0.79 mW	0.79 mW
Locking Time	1.9 μs	1.58 μs	1.93 μs	1.92 μs
Frequency	2.702 GHz	2.703 GHz	2.502 GHz	2.502 GHz

Bee Colony Optimization

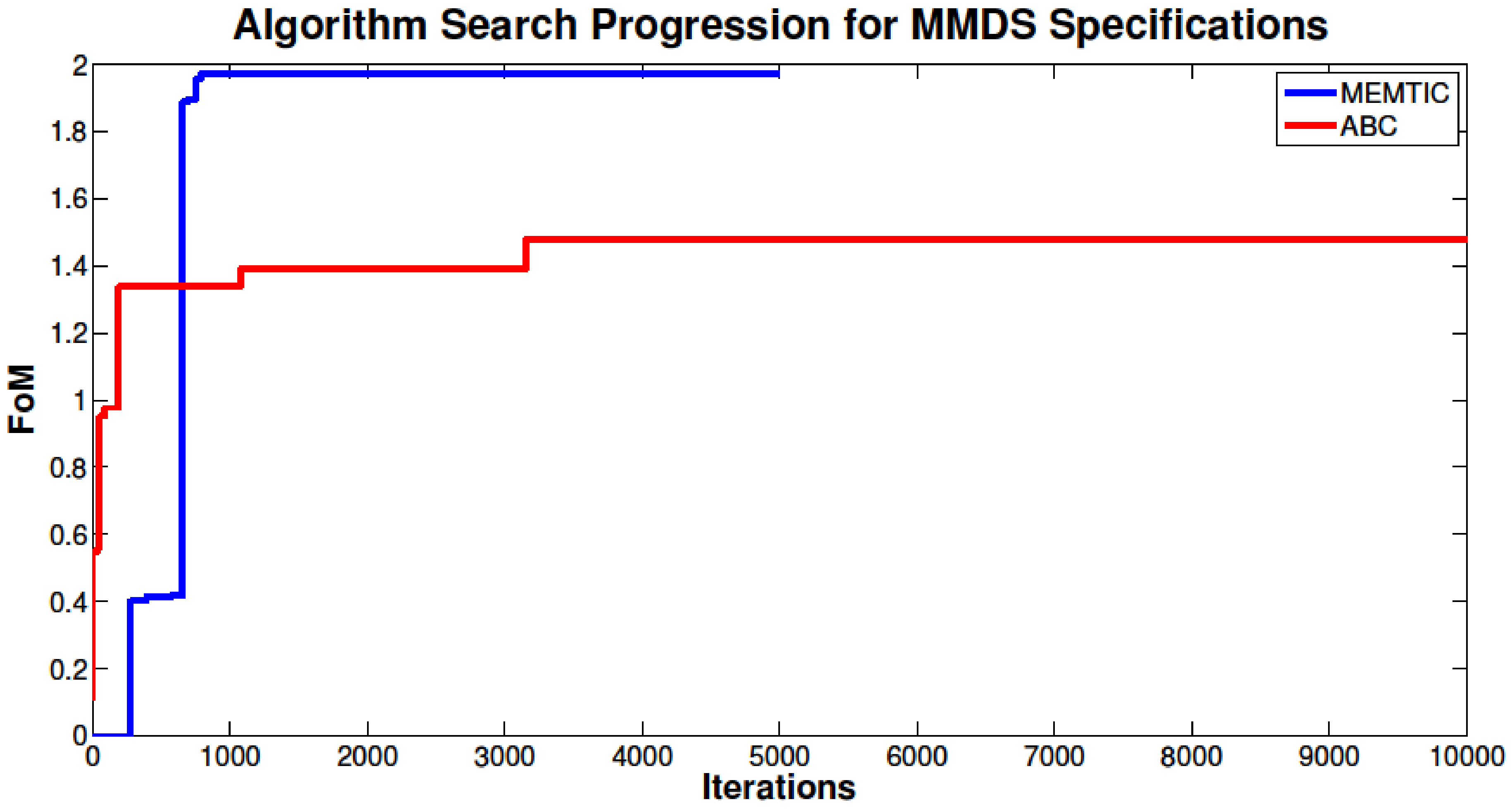


State diagram for bee transition

Algorithm 1 Proposed Bee Colony Optimization Algorithm.

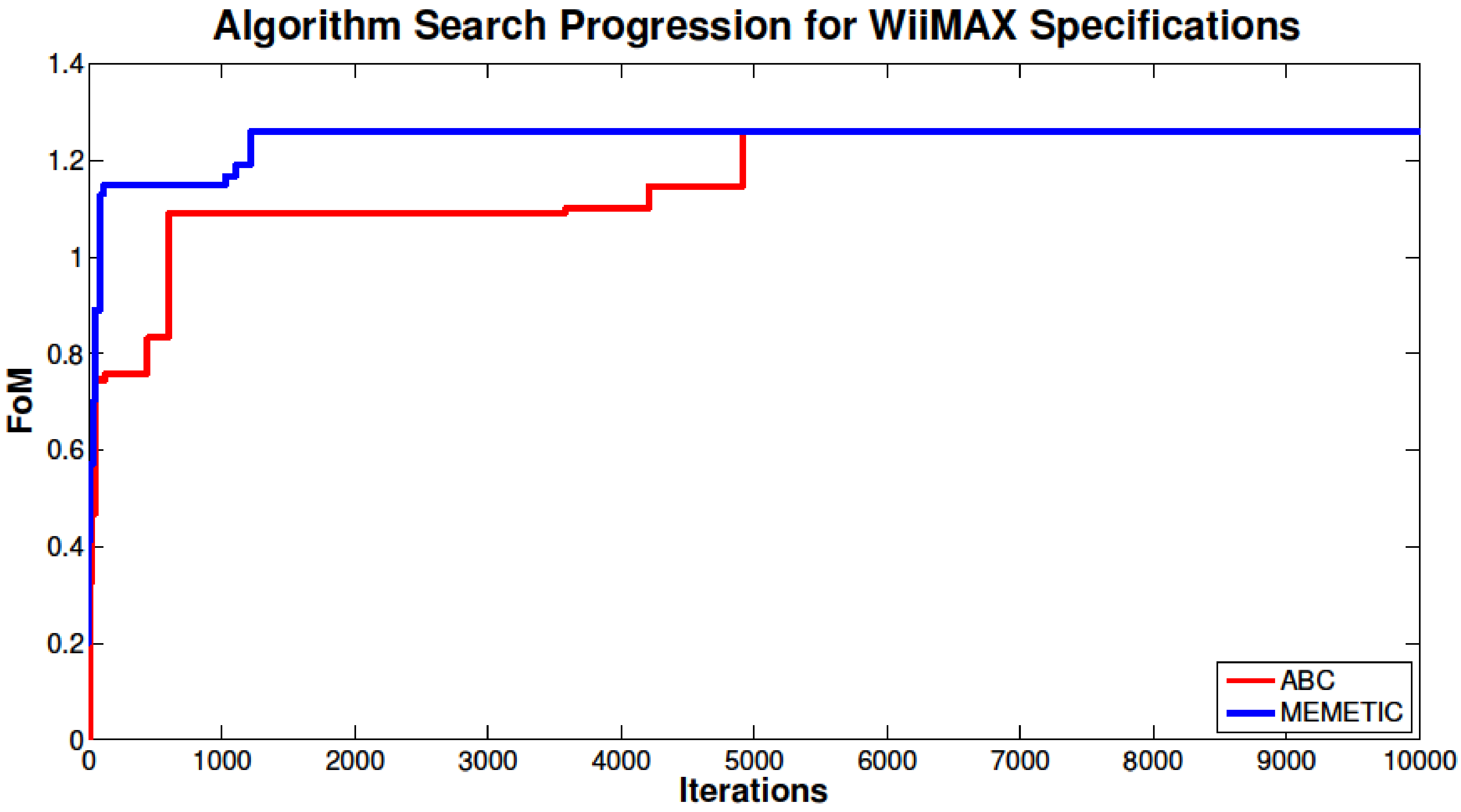
```
Initialize maximum iterations ← max_i.
Set the boundaries for each parameter P(i) ← [min, max].
NumberBees ← Define the number of bees.
buffer ← Number of close worker bees dispersal.
Initialize a matrix as follow: bee_matrix(3, NumberBees) ←
[workers, onlookers, scouts].
Set bee_matrix first half to be workers and other onlookers.
Initialize food sources.
while (Counter ≤ max_i) do
  for each i from 1 to NumberBees do
    if (bee_matrix(1, i) == 1) then
      (1) Send worker bee to a random known food source.
      Calculate Power(i), Jitter_h/v(i) using metamodels.
      Calculate the proposed FoM of the PLL.
      if (current FoM is better than the previous FoM) then
        Update result and location.
      else
        Convert bee to onlooker.
      end if
    else
      if (bee_matrix(1, i) == 1) then
        (2) Send onlooker bee.
        Calculate probability that the food source is good
        if (probability is high) then
          P ← (P_min + random(1) × P_max) × buffer.
          Send onlooker to random location for each design
          parameter P.
          Calculate the FoM.
          if (current FoM is better than the previous FoM)
          then
            Update result and location.
            Convert bee to worker.
          else
            Convert bee to scout.
          end if
        end if
      else
        (3) Send scout bee.
        Pick the best result as best_r.
        P ← P_min + random(1) × P_max.
        Send the scout to random location for each P.
        if (current FoM is better than the previous FoM) then
          Update the result.
          Convert bee to worker.
        end if
      end if
    end if
    if (current FoM is better than previous FoM) then
      Update result and location.
    end if
  end for
  Counter ← Counter + 1.
end while
Return result and location.
```

Results



Algorithm Search Progression for MMDS Specifications
MEMTIC
ABC
FoM
Iterations
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

Results



Algorithm Search Progression for WiiMAX Specifications
FoM
Iterations
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
ABC
MEMETIC

ABC vs. Memetic Algorithm

TABLE 7.13. Comparison of Algorithms for speed and convergence.

Algorithm	Simulation Time	Convergence (iter.)		Speedup
		WiiMAX	MMDS	
ABC	≈12 min	4914	3193	1x
Memetic	≈5 min	1221	825	2.4x

Conclusion: Due to faster convergence and better global optimization results Memetic algorithm is more fit for analog/AMS optimization.

Conclusions

- A design flow for using neural network metamodels is proposed.
- A 180 nm PLL is subjected to the proposed design flow.
- ABC and memetic optimization algorithms are implemented for circuit transistor sizing optimization.
- IP reuse has been presented in this work for 2 different specifications of PLL successfully completing design using once created metamodels.
- Memetic algorithm shows faster convergence and global optimization accuracy Artificial Bee Colony algorithm.