Optimum Layout of Multiplexer with Minimal Average Power based on IWO, Fuzzy-IWO, GA, and Fuzzy GA

Farshid Keivanian[1], Nasser Mehrshad[2], Seyed-Hamid Zahiri[3]

[1] **Department of Electrical and Computer Engineering, University of Birjand, Birjand, South Khorasan, Iran**
FarshidKeivanian@birjand.ac.ir

[2] **Department of Electrical and Computer Engineering, University of Birjand, Birjand, South Khorasan, Iran**
Nmehrshad@birjand.ac.ir

[3] **Department of Electrical and Computer Engineering, University of Birjand, Birjand, South Khorasan, Iran**
Shzahiri@yahoo.com

Abstract

This article is based on the application of heuristic algorithms to solve the optimum solution for a VLSI circuit. The idea is to find the optimum layout for a 2-to-1 multiplexer with minimal average power. The objective function is the average power of 2:1 MUX with four MOSFETs with different channel widths. They make a four dimensional space which is searched by search agents of algorithm. Motivated by the convergence of Invasive Weeds Optimization (IWO) and Genetic Algorithm (GA) and the link of MATLAB with HSPICE Software the optimized layout of 2:1 MUX is obtained. Based on IWO, Fuzzy-IWO, GA, Fuzzy-GA algorithms the best resulting of MUX layout in Static NMOS Logic in 0.18µm Technology with supply voltage of 5v has the average power consumption of 3.6 nW with Fuzzy-IWO.

Keywords: *Minimal Average Power, Multiplexer, Invasive Weeds Optimization (IWO), Fuzzy-IWO, GA, Fuzzy-GA***.**

1. Introduction

The Multiplexers ("MUX") are switching circuits that switch or route signals from input to output path. They are combinational circuits that are memory less as there is no signal feedback path [1]. This article describes the heuristic algorithms of Invasive Weed Optimization (IWO) which is firstly proposed by A. R. Mehrabian, C. Lucas [2], Fuzzy-IWO which is defined in this article, GA that its performance is tested over layout problems [3], and Fuzzy-GA which is defined for this study. This article employs all of them to find the optimum layout with the most appropriate channel widths in 2:1 MUX circuit with minimal average power of consumption.

2. Multiplexer in Static NMOS Logic

The logic style is the way how the logic function is constructed from a set of transistors. The logic style influences the speed, size, power dissipation, and wiring complexity of a circuit. The advantage of pass-transistor logic style in comparison with CMOS logic style is that one pass-transistor network (either NMOS or PMOS) is sufficient to perform the logic operation [4, 5]. There are several pass-transistor logic styles such as NMOS pass-transistor logic, double pass-transistor logic (DPL), CMOS transmission gate, LEAN integrated pass gate logic (LEAP) and pass transistor logic (PTL) that are considered to implement 2-to-1 multiplexer [6,7,8]. The static NMOS logic multiplexer is the optimum device level design which has characteristics of high speed with minimum power compared with the other styles [9]. The Multiplexer in static NMOS pass-transistor logic uses two NMOS transistors as pass transistors that select input signals (x or y) to propagate to output (f). The symbol of 2:1 MUX is shown in figure 1, also its transistor design level is shown in figure 2. The 2:1 MUX with just two pass-transistors without any inverter is not used for this article since it has only two variables of channel widths W1 and W2. Since the search space for heuristic algorithms is better to be big enough otherwise the convergence will occur too soon and the search space of decision variables will not effectively searched. The threshold voltage of both NMOS pass-transistors in figure 2 should be identical for accurate operation [9]. Also the equation 1 shows that the threshold voltage (V_t) does not depend on channel widths (W) [10] therefore the heuristic algorithms can be employed to search several different channel widths in a 4-dimensional search space to find the best solution [W_1 W_2 W_3 W_4] without any problem.

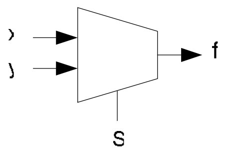

Fig. 1 The Graphical Symbol of 2:1 Multiplexer

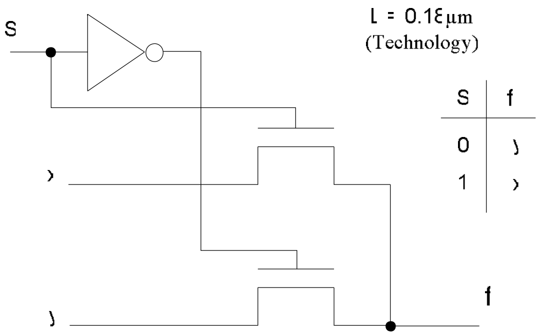

Fig. 2 NMOS pass-transistor logic in 2:1 Multiplexer

$$V_t = \frac{\sqrt{2qN_A\varepsilon\,(2\varphi_f)}}{C_{ox}} + 2\varphi_f + \varphi_{ms} - \frac{Q_{ss}}{C_{ox}} \qquad (1)$$

3. Heuristic Algorithms: IWO and GA

Nowadays computers are used to solve incredibly complex problems. Usually heuristic algorithms are developed to have low time complexity and applied to the complex problems. The heuristic algorithms suggest some approximations to the solutions of optimization problems. In optimization problems the objective is to find the optimal of all possible solutions, that minimize or maximize the objective function. In this article it is minimization kind. The objective function is one used to evaluate a quality of the generated solution. The collection of all possible solutions for a given problem can be regarded as a search space, and the optimization algorithm is referred as search algorithm [11].

3.1 The Heuristic Algorithm of Invasive Weed Optimization (IWO)

IWO is an ecologically inspired stochastic optimization algorithm which has shown successful results for global optimization [12]. There has been several articles for the application of IWO in electrical engineering and their conclusion was about the acceptable performance of IWO[13]. This algorithm is inspired by the growth of weeds. Weeds are plants that their growth is the significant threat to crop plants. They are so sustained and adapt to the changing environment. So based on characteristics of weeds and simulation their characteristics a powerful algorithm of IWO can be achieved [14]. Invasive weeds grow in an area globally and they can not be removed or controlled by human. The more energy the farmers spend to destroy them, the better or more they become, so the invasive weeds always win. The reasons for this claim are that a) after thousand years of farming there is still invasive weeds. b) they have existence even after the application of herbicides for them c) the appearance of new species of weeds is widely on earth d) they adapt to the environment. The treat of invasive weeds in occupying the territories and establish colonies is respectively as following steps:

1) create opportunities for spaces of pruning system. 2) weed infestations in areas of significant opportunity to establish colonies and occupied space. 3) create various plants in order to exploit the opportunities to exploit the

spaces (biodiversity of weeds). In a simulation of the behavior of weeds there are some steps: Step I) the spread of seeds in the desired space where they can change to weeds and the creation of initial population occurs which is random and it is shown in a flowchart in figure 3. Step II) Each member of weeds can has offspring and produce seeds around it based on its merit or cost function value. The more merit weeds are the more offspring occurs around them. The process of offspring in step II is shown in figure 4. The number of seeds that are produced around each weed is limited to the range of [Smin Smax]. And S_{max} shows the maximum number of seeds that have offspring around the weed which has the least cost value. Whatever the merit of the weed is more the more seed around it will be produced and this relation is linear which is illustrated in figure 4. The number of seeds which are produced around each weed is based on the equation 2.

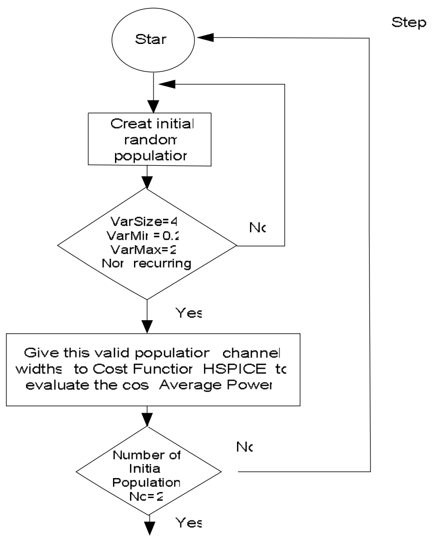

Fig. 3 Step I: Creation of initial random population

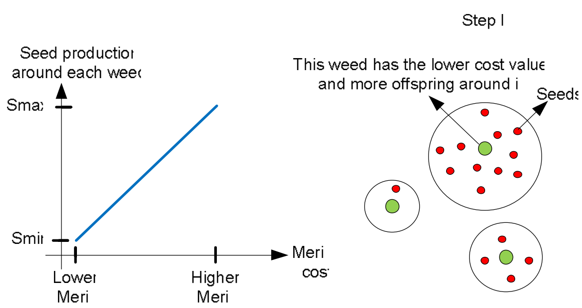

Fig. 4 Step II: The process of offspring

$$Seed_i = Round\{S_{\min} + (S_{\max} - S_{\min}) \times \frac{N_{weed} - \mathrm{rank}_i}{N_{weed} - 1}\} \tag{2}$$

$Seed_i$ is the number of produced seeds around $weed_i$. As the number of seeds does not have any decimal point, the operator of "Round" is used in relation 2. S_{min} is the minimum number of seeds that can be produced around each weed. In contrast with that, S_{max} is the maximum number of seeds around each weed. N_{weed} is the initial population of weeds. $rank_i$ is the rank of $weed_i$. If the rank is the highest, ("$rank_i$ = 1") the numerator and denominator can be simplified and finally $Seed_i = S_{max}$ also if $rank_i = N_{weed}$ then $Seed_i = S_{min}$. When the cost value of $weed_i$ is the highest value then the minimum number of seeds will be created around it. These IWO parameters are set as S_{min} =1 and S_{max}=10. When S_{min} = 1 then there is a chance for a bad weed with high cost value to have offspring to be in better position and cost while if S_{min} is set to 0 then no seed will be spread around a bad weed with high cost function value.

The amount of standard deviation (SD) in each iteration of IWO algorithm is updated. The SD of seeds around each weed becomes lower when reach to the end of algorithm while it is high at the first of algorithm. The produced seeds in search space will be spread around weeds with normal distribution with zero mean and a variance. The standard deviation is changed in each iteration of algorithm and it will start from initial standard deviation ($\delta_{initial}$) and decrease to the final standard deviation (δ_{final}) in the range of [$\delta_{initial}$=0.3 , δ_{final}=0.001] , this change is non-linear and based on equation 3.

$$\delta_{Iter_i} = \frac{(MaxIt - \mathrm{Iter}_i)^n \times (\delta_{initial} - \delta_{final})}{(MaxIt - 1)^n} + \delta_{final} \tag{3}$$

δ_{Iteri} is the standard deviation of i iteration since the SD is updated in each iteration. MaxIt is the maximum number of iteration. $Iter_i$ is the i iteration. The next parameter is n which is nonlinear coefficient. modulation index in this article is set to 3 ("n=3") which is common in algorithms. The SD decreases with increasing iteration. This concept is the same as the statement of update standard deviation in pseudo code in Box 1. The concept of standard deviation is graphically shown in circles with decreasing radius in figure 5. Step III) Another process in IWO is competitive removal which is referred in pseudo code in Box 1. The predetermined amount of weeds P_{max} is determined at the first of the algorithm. If the number of weeds and seeds become more than it then the removal is done based on merit. The population is sorted based on objective function of average power in this article. Those who have lower average power will move to the next generation and the others will be removed from the population. In fact, the layouts will be sorted based on their average power and

P $_{max}$ number of layouts will be selected and others with higher average power will be deleted in this article. The three steps (Step I, II, and III) all are shown in a flowchart in figure 6. The step III is inspired by this fact that growing and existence of weeds depends on their desirability.

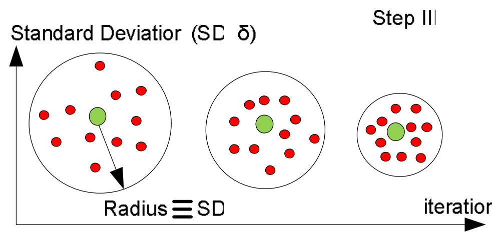

Fig. 5 Step III: The decreasing of standard deviation with increasing iteration

The pseudo code of Invasive Weed Optimization (IWO) is shown in Box 1 also the flowchart of IWO is illustrated in figure 6.

Box 1. IWO Pseudo-Code in this Article

- Set the IWO parameters VarMin=0.2, VarMax=2, nVar=4, MaxIt=10, N_{weed}=2, P_{max}=4, S_{min}=1, S_{max}=10, n=3, $\delta_{initial}$=0.3, δ_{final}=0.001
- **Do while** population < N_{weed}
  Create initial random population based on (VarMin, VarMax, VarSize)
  Call Objective Function and save the cost function for each population
- **End Do**
  Sort the population based on their cost function
  - **Do While** iteration number (it) < Maximum Iteration (MaxIt)
    - Production of seeds with normal distribution with mean=position and the SD which is updated in each iteration around each weed based on the cost function and update the SD for each weed and check the range of variable [VarMin , VarMax]
    - Sort the solutions based on their objective functions and remove the extra individuals more than Pmax (Competitive Removal)
    - Save the results (BestCost)
  - **End Do**
- Plot Results

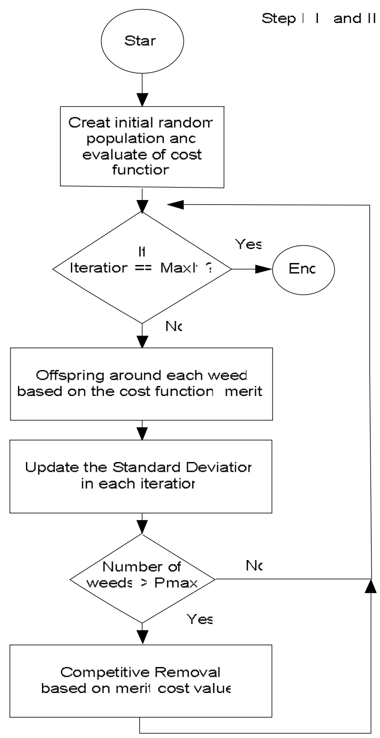

Fig. 6 The flowchart of IWO algorithm (Step I, II, and III)

The search agents in IWO are named weeds, as in Genetic Algorithm (GA) they are chromosomes. The problem which is defined in IWO is the optimization of layout in 2:1 multiplexer with minimal average power which is illustrated in figure 7. The average power of a 2:1 multiplexer in NMOS static logic in technology 0.18µm (L=0.18) is calculated by HSPICE in each iteration of algorithm. The objective function pseudo code is in Box 2.

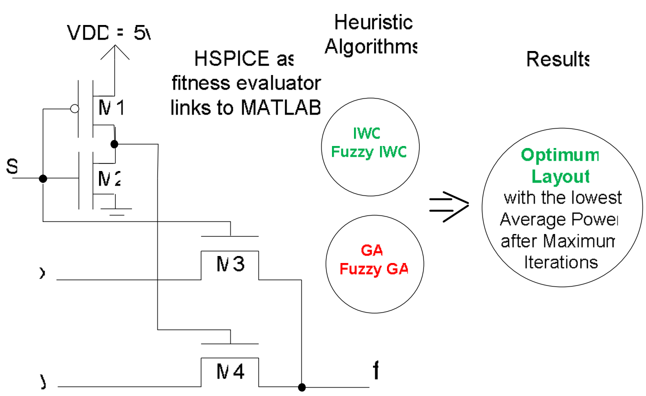

Fig. 7 The link between MATLAB and HSPICE for the optimization of Layout in 2:1 MUX with minimal average power

Box 2. The Pseudo-Code of Objective Function which is defined in this Article for IWO and GA Algorithms

- **O**pen the input netlist file ('C3.sp')
- **S**eek the MOSFETs and save their channel widths as positions ("Individual.Positions") for IWO main program
- **R**un HSPICE as a fitness evaluator with input Netlist File ('C3.sp')
- **R**ead data from output file ('C3.lis')
- **S**eek the average power ('avgpower') from the output file ('C3.lis') and save it for IWO main program

In this article the number of MOSFETs is 4 so the number of decision variables in MATLAB is 4 (nVar=4). Also IWO algorithm is continuous kind thus each solution is continuous. The results (W_1 …W_4) are continuous values between VarMin=0.2μm and VarMax=2μm. About IWO parameters , after some implementations of algorithm in MATLAB, the non-practical result of figure 8 is obtained while N_{weed}=50, P_{max}=100, and MaxIt=25. Because the number of variables is very low (nVar=4), the search space is so small and there is no need to set the high values for IWO parameters. With the above values the cost function of average power converges to 0 which is not practical for an electronic circuit. After that, again, the parameters of IWO are set as N_{weed} = 2, Pmax = 4, MaxIt = 10 and figure 9 is obtained. It is recommended that the maximum number of weed population, P_{max}, is defined as twice the initial population N_{weed} [14].

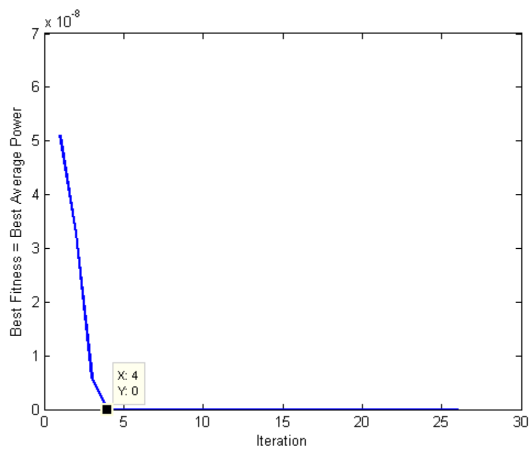

Fig. 8 Non-practical result of average power by setting high values for IWO parameters : N_{weed}=50, P_{max}=100, Maxit=25

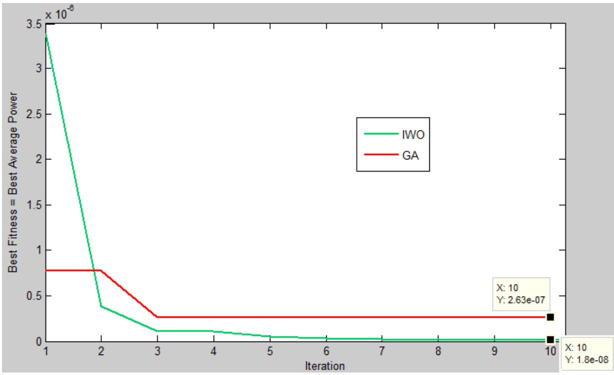

Fig. 9 The practical result of average power by setting lower values for IWO parameters : N_{weed}=2, P_{max}=4, Maxit=10, the Average Power result based on IWO is 1.8*e-8 watt in green color, and it is 2.63*e-7 watt based on GA in red color

The best solutions of variables for IWO and GA algorithm are respectively shown in table 1 and 2.

Table 1: The Best Solutions (Channel Widths) based on IWO Algorithm

W1	***W2***	***W3***	***W4***
1.804673 48507489	0.46345533 5930802	0.4445454 72046924	0.8365723 96678707

Table 2: The Best Solutions (Channel Widths) based on GA Algorithm

W1	***W2***	***W3***	***W4***
0.200000 00000000 0	0.94614057 0007595	1.2113924 9682907	1.5751227 4128552

3.2 Fuzzy-IWO

In 3.1 part, based on equation 3, the standard deviation, SD decreases with increasing the iteration and the search step becomes so fine. Now if a local solution occurs at the end of the algorithm when the search step is fine and SD is low the algorithm of IWO will focus on a solution with high cost function value. So it is better that the control of parameter Sigma, SD becomes fuzzy. This means that the amount of Sigma would better to be dependent on the number of iteration and cost function. At the first of search when the iteration is low and cost function is high, sigma should be big enough, and at the end of search that the iteration is high and the cost function value is high then the parameter of sigma should be high to increase the diversity of search and increment the search step. The fuzzy rules for Fuzzy-IWO are shown in table 3. The Fuzzy Inference System (FIS) used for IWO is shown in figure 10. The type of membership function which is used for all FIS variables (itnormalized, BestCostnormalized, and Sigma) is trapmf. The normalization of variables is done based on relation 4 and 5.

Table 3: The Fuzzy Rules in Fuzzy-IWO

If	***Then***
Itnormalized is High and BestCostnormalized is High	Sigma is High
Itnormalized is High and BestCostnormalized is Low	Sigma is Low
Itnormalized is Low and BestCostnormalized is High	Sigma is High
Itnormalized is Medium and BestCostnormalized is Medium	Sigma is Medium

$$Itnormalized = \frac{it}{MaxIt} \qquad MaxIt = 10, it:1...MaxIt \tag{4}$$

$$BestCostnormalized = \frac{WorstCost(it) - BestCost(it)}{WorstCost(it)} \quad MaxIt = 10, it:1...MaxIt \tag{5}$$

The normalization is done to convert the values of iteration and best cost between zero and one. The FIS system works parallel with IWO algorithm is shown in figure 10.

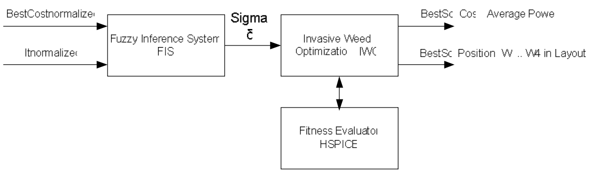

Fig. 10 The Fuzzy Inference System used in this article for Fuzzy-IWO

Also the pseudo code for fuzzy-IWO is shown in Box 3.

Box 3. Fuzzy-IWO Pseudo-Code in this Article

- **S**et the IWO parameters VarMin=0.2, VarMax=2, nVar=4, MaxIt=10, N_{weed}=2, P_{max}=4, S_{min}=1, S_{max}=10, n=3, δ=0.3
- **Do while** population < N_{weed}
  Create initial random population based on (VarMin, VarMax, VarSize)
  Call Objective Function and save the cost function for each population
- **End Do**
  Sort the population based on their cost function
  - **Do While** iteration number (it) < Maximum Iteration (MaxIt)
    -**P**roduction of seeds with normal distribution with mean=position and the SD ('δ') is determined based on fuzzy rules and spreading them around each weed based on the cost function within [VarMin , VarMax]
    -**S**ort the solutions based on their objective functions and remove the extra individuals more than Pmax (Competitive Removal)
    -**N**ormalization of variables : itnormalized = it / Maxit, and BestCostnormalized = [WorstCost(it) - BestCost(it)] / WorstCost(it)
    -**R**ead Fuzzy Inference System File and Fuzzy Rules (Fuzzy_IWO_FIS.fis)
    -**S**ave the results (BestCost)
  - **End Do**
- **P**lot Results

3.3 GA Algorithm

Genetic Algorithm has been used to optimize electronics applications such as combinational circuits at the gate level in several papers [15,16]. In this article the goal is the optimization at the layout level in VLSI circuit of 2:1 MUX. In Genetic Algorithm the search agents are the chromosomes while they are defined as weeds in IWO in previous part. In GA the best chromosomes have the best objective function. Based on pseudo in Box 4 the program worked and the least average power which obtained by GA was compared in figure 9 with IWO result.

Box 4. GA Pseudo-Code in this Article

- **D**efine the problem of Multiplexer 2:1, Cost Function=Objective Function ("in Box1"), nVar=4, VarMin=0.2, VarMax=2, MaxIt=10, nPop=4 (Population Size) , pc=0.8 (Crossover Percentage), pm=0.3 (Mutation Percentage)
- **G**enerate Initial Population, Evaluate, Sort, and place them in the Best Solutions
  - **Do While** iteration number (it) < Maximum Iteration (MaxIt)
    - **C**alculate Selection Probabilities
    - **S**elect Parents indices and apply crossover and evaluate off springs, in Crossover Operator
    - **S**elect Parents indices and apply mutation and evaluate Mutant, in Mutation Operator
    - **C**reate Merged Population, sort population
    - **U**pdate and store the best solution and Cost that ever found
  End Do
Plot Results

3.4 Fuzzy-GA Algorithm

The fuzzy inference system which is used with Genetic Algorithm is shown in figure 11. The pseudo code for Fuzzy-GA is shown in Box 5.

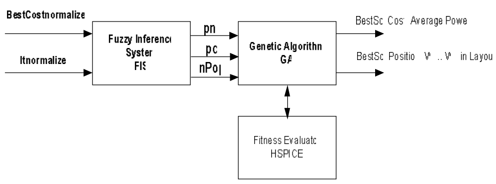

Fig. 11 The Fuzzy Inference System in this article for Fuzzy-GA

Box 5. Fuzzy-GA Pseudo-Code in this article

- **D**efine the problem of Multiplexer 2:1, Cost Function=Objective Function ("in Box1"), nVar=4, VarMin=0.2, VarMax=2, MaxIt=10, nPop=4 (Population Size) , pc=0.8 (Crossover Percentage), pm=0.3 (Mutation Percentage)
- **G**enerate Initial Population, Evaluate, Sort, and place them in the Best Solutions
  - **Do While** iteration number (it) < Maximum Iteration (MaxIt)
    - **C**alculate Selection Probabilities
    - **S**elect Parents indices and apply crossover and evaluate off springs,

in Crossover Operator

- Select Parents indices and apply mutation and evaluate Mutant, in Mutation Operator
- Create Merged Population, sort population
- Update the best solution and the worst solution that ever found and Store Best Cost and Worst Cost
- Normalization of variables : itnormalized = it / MaxIt, and BestCostnormalized = [WorstCost - BestCost(it)] / WorstCost
- Read Fuzzy Inference System File and Fuzzy Rules (Fuzzy_GA_FIS.fis)
- Save the results (BestCost)

- End Do

- Plot Results

The optimum control of mutation percentage (“pm”), crossover percentage (“pc”), and the number of population (“nPop”) is done in Fuzzy-GA. The Fuzzy-GA rules are illustrated in table 4.

Table 4: The Fuzzy Rules in Fuzzy-GA

If	*Then*
Itnormalized is High and BestCostnormalized is High	pm is High and pc is Low and nPop is High
Itnormalized is High and BestCostnormalized is Low	pm is Low and pc is High and nPop is Low
Itnormalized is Low and BestCostnormalized is High	pm is High and pc is Medium and nPop is High
Itnormalized is Medium and BestCostnormalized is Medium	All pm, pc, and nPop are Medium

By implementation of Fuzzy-GA the best average power converges to $5.21*10^{-7}$ watt while with Fuzzy-IWO the best average power is $3.6*10^{-9}$ watt which is much better than Fuzzy-GA. The best average power based on both Fuzzy-GA and Fuzzy-IWO are illustrated in figure 12.

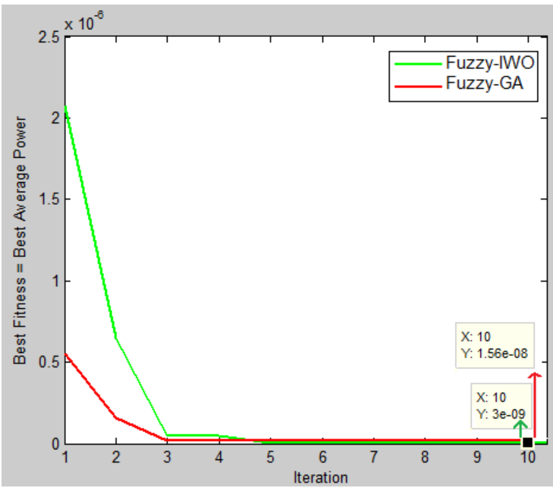

Fig. 12 The best average power based on Fuzzy-IWO is 3.6*e-9 watt in green color and it is 5.21*e-7 watt based on Fuzzy-GA in red color

The Positions of the best solutions in Fuzzy-IWO are shown in table 5. Also the best solutions by Fuzzy-GA are shown in table 6 finally the comparison is in the next part.

Table 5: The Best Solutions ('BestSol.Position') or (channel widths) based on Fuzzy-IWO

W1	*W2*	*W3*	*W4*
0.200000 00000000 0	0.94614057 0007595	1.2113924 9682907	1.5751227 4128552

Table 6: The Best Solutions (Channel Widths) based on Fuzzy-GA

W1	*W2*	*W3*	*W4*
0.219542 9318496	0.4251715 87415530	1.612277 04853422	0.253744 32945932

4. Conclusions

By implementation of heuristic algorithms such as IWO, Fuzzy-IWO, GA, Fuzzy-GA the optimum 2-to-1 multiplexer is obtained. The best layout consumes only 3nW based on Fuzzy-IWO. The comparisons are illustrated in table 7 and figure 13. Also the fuzzy-GA and fuzzy-IWO showed the better results since the fuzzy rules has increased the performance of algorithm and finally the most optimum 2:1 MUX in NMOS logic with 2 pass-transistors and one inverter in technology L=0.18µm and V_{DD}=5v is found in search space of four independent variables W_1 W_2 W_3 W_4 . The least average power consumption is obtained by Fuzzy-IWO.

Table 7: The comparison of average power (Best Solutions.Cost)

Heuristic Algorithms	*The best fitness value or the lowest average power which is obtained*
IWO	18 nW
Fuzzy-IWO	3 nW

GA	263 nW
Fuzzy-GA	15.6nW

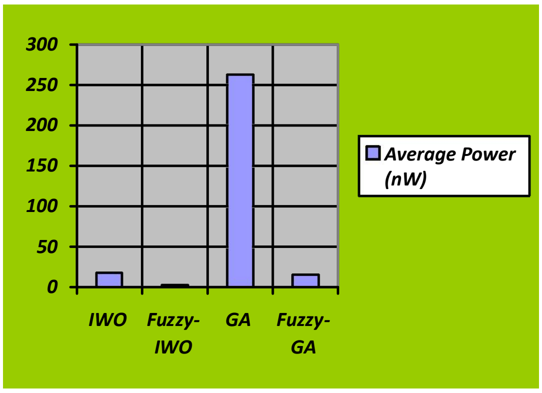

Fig. 13 The Comparison of the best average powers (the least objective functions) for IWO, Fuzzy-IWO, GA, and Fuzzy-GA.

Acknowledgments

Thanks Professor Seyed-hamid Zahiri and Dr. A. Bijari for all help, also thanks my wife and parents for their patience.

References

[1] Wayne Storr, Tutorial Website of Electronics, Copyright © 1999 – 2014

[2] A. R. Mehrabian, C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization", International Journal of Ecological Informatics, Elsevier, Volume 1, Issue 4, Pages 355– 366, December 2006.

[3] I. Mihajlovic , Z. Zivkovic, N. Strbac, D. Zivkovic and A. Jovanovic, "USING GENETIC ALGORITHMS TO RESOLVE FACILITY LAYOUT PROBLEM", Serbian Journal of Management 2 (1) 35-46 , 2007

[4] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, Top-down pass-transistor logic design, IEEE Journal of Solid-State Circuits, Vol. 31, June 1996, 792-803.

[5] Radhakrishnan D. Fabricius, Introduction to VLSI design, (Tata McGraw-Hill edition, New Delhi, 2005)

[6] Jan M Rabaey, Anantha Chandrakasan, Borivoje Nikolic, Digital Integrated Circuits: A Design perspective, (2nd Edition, Prentice-Hall Inc).

[7] Stephen brown, Zvonko Vranesic, Fundamentals of Digital logic with VHDL design, (Tata McGraw-Hill, new Delhi, 2002).

[8] N. Ohkubo et al. , A 4.4 ns CMOS 54x54 multiplier using pass transistor multiplexer, IEEE Journal of Solid-State Circuits, vol. 30, March 1995, 251-257.

[9] M. Padmaja, V.N.V. Satya Prakash, "Design of a Multiplexer in Multiple Logic Styles for Low Power VLSI", International Journal of Computer Trends and Technology- Volume 3, Issue 3, 2012.

[10] Paul. R. Gray, Paul J. hurst, Stephen H. Lewis, Robert G. Meyer, "Analysis and Design of Analog Integrated Circuits" , Chapter 1 (1.167 formula, 1.5.4 part , page 47), 5th Edition, ISBN 978-0-470-24599-6, Book, Copyright 2009 © John Wiley & Sons, Inc.

[11] Natallia Kokash, "An introduction to heuristic algorithms", thesis in Department of Informatics and Telecommunications, University of Trento, Italy

[12] Amir Hossein Nikoofard, Hossein Hajimirsadeghi, Ashkan Rahimi-Kian, Caro Lucas, "Multiobjective invasive weed optimization: Application to analysis of Pareto improvement models in electricity markets" , International Journal of Applied Soft Computing, ELSEVIER, Volume 12, Issue 1, January 2012, Pages 100-112.

[13] Ashik Ahmed, B.M. Ruhul Amin, "Performance Comparison of Invasive Weed Optimization and Particle Swarm Optimization Algorithm for the tuning of Power System Stabilizer in Multi-machine Power System", International Journal of Computer Applications, 0975-8887, Volume 41-No.16, March 2012.

[14] Meysam Rabie, "Theoretical and Practical training of Invasive Weeds Optimization Algorithm, IWO" , Faculty member of Bu-Ali Sina University, Engineering College of Tuyserkan , Iran, The training source of Artificial Intelligence in Iran; Tutorial website of www.matlabsite.com

[15] Arturo Hernandez Aguirre, Carlos A. Coello Coello, "Using Genetic Programming and Multiplexers for the synthesis of logic circuits", Journal of Engineering Optimization, 2004.

[16] Carlos A.Coello Coello, Alan D. Christiansen, and Arturo Hernandez, "Automated Design of Combinational Logic Circuits using Genetic Algorithms", Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, ICANNGA'97. University of East Anglia, Norwich, England. Edited by D. G. Smith, N. C. Steele and R. F. Albrecht, Springer-Verlag, pp. 335-338. 2-4 April 1997.

Farshid Keivanian Master of Science in Electronics Engineering at University of Birjand 2013-2015 ,Iran BSc in Electronics Engineering 2001-2005, Articles by now: "Optimization of JK Flip Flop Layout with Minimal Average Power of Consumption based on ACOR, Fuzzy-ACOR, GA, and Fuzzy-GA" in review process for publication in International Journal of Mathematics and Computer Science 2014, ISSN: 2008-949X, Poland.

Nasser Mehrshad Associate Professor of electrical engineering at the University of Birjand, Iran. Research interests: Signal Processing, Image Processing, Modeling the human visual system, Biomedical

Instrumentation.

Seyed-Hamid Zahiri Professor of electrical engineering , Department of Electrical Engineering, Faculty of Engineering, Birjand University, Birjand, Iran. Research interests: Evolutionary Optimization, Optimization of Swarm Intelligence, Soft Computing, Pattern Recognition, Image Processing.