# Adaptive Crossover and Mutation in Genetic Algorithms Based on Clustering Technique


Jun ZHANG
Dept. of Computer Science
SUN Yat-sen University
Guangzhou, China
86-84114898
junzhang@ieee.org

Henry, S.H. Chung
(Corresponding author)
Dept. of Electronic Engineering
City University of Hong Kong
852-27887807
eeshc@cityu.edu.hk

Jinghui Zhong
Dept. of Computer Science
SUN Yat-sen University
Guangzhou, China
86-84114898
issai@zsu.edu.cn



## ABSTRACT

Instead of having fixed $p_x$ and $p_m$, this paper presents the use of fuzzy logic to adaptively tune $p_x$ and $p_m$ for optimization of power electronic circuits throughout the process. By applying the *K*-means algorithm, distribution of the population in the search space is clustered in each training generation. Inferences of $p_x$ and $p_m$ are performed by a fuzzy-based system that fuzzifies the relative sizes of the clusters containing the best and worst chromosomes. The proposed adaptation method is applied to optimize a buck regulator that requires satisfying some static and dynamic requirements. The optimized circuit component values, the regulator's performance, and the convergence rate in the training are favorably compared with the GA's using fixed $p_x$ and $p_m$



## Categories and Subject Descriptors

D.2.2 [**Evolutionary prototyping**]

## Keywords

Genetic Algorithms and Real World Applications


## 1. INTRODUCTION

This paper presents the use of fuzzy logic to adaptively tune $p_x$ and $p_m$ for optimization of PEC throughout the process. By applying the *K*-means algorithm[1] , distribution of the population in the search space is clustered in each training generation. Inference of $p_x$ and $p_m$ is performed by a fuzzy-based system that fuzzifies the relative sizes of the clusters containing the best and worst chromosomes. Both of the population distribution factor and the computational efficiency, as compared with[2] and[3], are considered. The proposed adaptation method is applied to optimize a buck regulator that requires satisfying some static and dynamic requirements. The decoupled optimization technique as proposed in[4] is used. Nevertheless, without loss of generality, the proposed parameter adaptation scheme can be applied to other GA-based optimization problems. The optimized circuit component values, the regulator's performance, and the convergence rate in the training are favorably compared with the GA's using fixed $p_x$ and $p_m$.

## 2. ADAPTIVE CONTROL OF $p_x$ AND $p_m$

Biological evolution shows that $p_x$ and $p_m$ should be adapted and should depend on the evolution state[5]. Thus, in order to enhance the training efficiency of[4], an adaptive approach for tuning $p_x$ and $p_m$ is proposed. The basic concept is based on considering that $p_x$ determines the probability of reproduction from parent chromosomes and $p_m$ determines the probability of creation from a parent chromosome in different training states. Fig. 1 illustrates the strategy of tuning $p_x$ and $p_m$ in four optimization states, including initial state, under-matured state, maturing state, and matured state [5]. In order to prevent premature convergence of the GA to a local optimum, it is essential to be able to identify whether the GA is converging to an optimum. The proposed method suggests the use of the relative population distribution to define the training state. The first step is to partition the population into clusters. Chromosomes of having similar component vectors are grouped in the same cluster. The second step is to use a fuzzy system that fuzzifies the relative sizes of the clusters containing the best and worst chromosomes to determine $p_x$ and $p_m$. The procedures are described as follows.

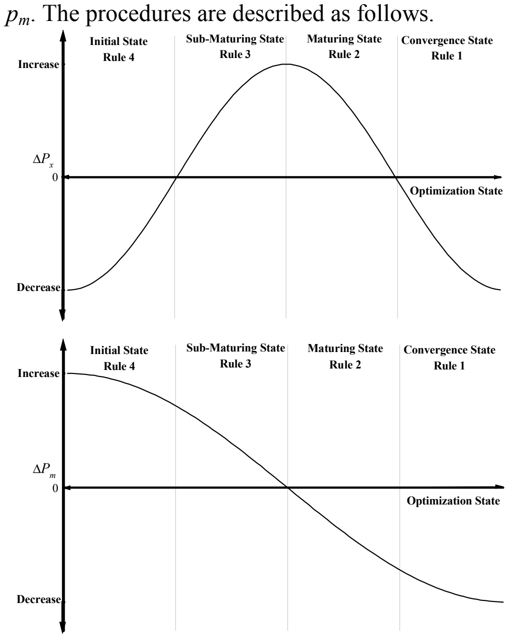


**Fig 1**. Illustrations on adjusting $p_x$ and $p_m$ in different optimization phases.

*A. Clustering of the Population*

Although *K*-means algorithm can only partition sub-optimal clusters, it is sufficient for this particular application to depict the chromosome distribution.



*GECCO'05,* June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

*B.Tuning Rules for $p_x$ and $p_m$*

Tuning of $p_x$ and $p_m$ in the proposed fuzzy inference system is based on considering the relative cluster sizes of $G_B$ and $G_W$ (i.e., $\hat{G}_B$ and $\hat{G}_W$ ). The following four rules for tuning $p_x$ and $p_m$ are defined and are tabulated in Table I.

Table I Strategy in tuning $p_x$ and $p_m$

| Size of the cluster containing the WORST chromosome | | | |
|---|---|---|---|
| | Large | $P_x$ Decrease<br>$P_m$ Increase | $P_x$ Increase<br>$P_m$ Decrease |
| | Small | $P_x$ Increase<br>$P_m$ Increase | $P_x$ Decrease<br>$P_m$ Decrease |
| | | Small | Large |
| | | Size of cluster containing the BEST chromosome | |

*Rule1-The best chromosome is in the largest cluster whilst the worst chromosome is in the smallest cluster.*

*Rule2-$G_B$ equals $G_W$. Both of them are the largest among others.*

*Rule3-$G_B$ equals $G_W$. Both of them are the smallest among others.*

*Rule4-The best chromosome is in the smallest cluster whilst the worst chromosome is in the largest cluster.*

*C.Fuzzy-based tuning mechanism for $p_x$ and $p_m$*

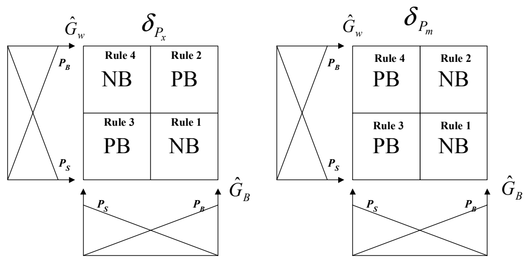


**Fig 2**. Illustrations on adjusting $p_x$ and $p_m$ in different optimization phases

Table II Fuzzy control rules for tuning $p_x$ and $p_m$

| Rule for $\delta_{P_x}$ |
|---|
| *Rule 1=(Rule (0,1)): If (* $\hat{G}_B$ *is $P_B$) and (* $\hat{G}_W$ *is $P_S$) then $\delta P_x$=NB* |
| *Rule 2=(Rule (1,1)): If (* $\hat{G}_B$ *is $P_B$) and (* $\hat{G}_W$ *is $P_B$) then $\delta P_x$=PB* |
| *Rule 3=(Rule (0,0)): If (* $\hat{G}_B$ *is $P_S$) and (* $\hat{G}_W$ *is $P_S$) then $\delta P_x$=PB* |
| *Rule 4=(Rule (1,0)): If (* $\hat{G}_B$ *is $P_S$) and (* $\hat{G}_W$ *is $P_B$) then $\delta P_x$=NB* |
| Rule for $\delta_{P_m}$ |
| *Rule 1=(Rule (0,1)): If (* $\hat{G}_B$ *is $P_B$) and (* $\hat{G}_W$ *is $P_S$) then $\delta P_m$=NB* |
| *Rule 2=(Rule (1,1)): If (* $\hat{G}_B$ *is $P_B$) and (* $\hat{G}_W$ *is $P_B$) then $\delta P_m$=NB* |
| *Rule 3=(Rule (0,0)): If (* $\hat{G}_B$ *is $P_S$) and (* $\hat{G}_W$ *is $P_S$) then $\delta P_m$ =PB* |
| *Rule 4=(Rule (1,0)): If (* $\hat{G}_B$ *is $P_S$) and (* $\hat{G}_W$ *is $P_B$) then $\delta P_m$=PB* |

# 3. DESIGN EXAMPLE & COMPARISONS

The proposed method is illustrated with the same example in [4]. The circuit schematic is shown in Fig. 3. The PCS is a classical buck converter and the FN is a proportional-plus-integral controller. In[4], $p_x$ (= 0.85) and $p_m$ (= 0.25) are fixed in the GA's. Fig. 4 shows the comparisons of the fitness values against the training generations with the fixed and proposed fuzzy-controlled $p_x$ and $p_m$. It can be seen that the fuzzy-controlled scheme can significantly improve the fitness values.

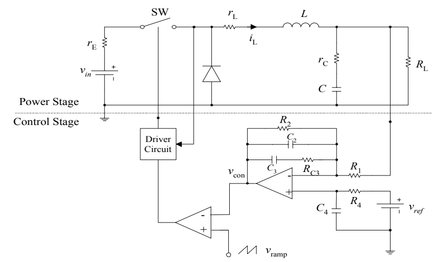


**Fig. 3**. Schematics of the buck regulator in[4]

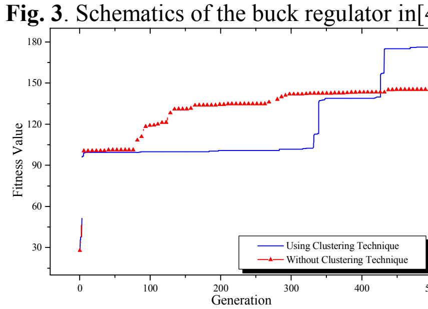


**Fig.4** Comparisons of the fitness values against the training generation using fixed and fuzzy-controlled $p_x$ and $p_m$

# 4. CONCLUSIONS

A fuzzy-controlled crossover and mutation probabilities in GA's for optimization of PECs has been proposed. They are determined adaptively for each solution of the population. It is in the manner that the probabilities are adapted to the population distribution of the solutions. This not only improves the convergence rate of the GA, but also prevents the GA from getting stuck at a local minimum. A buck regulator has been optimized. The results are favorably compared with the ones using GA's with fixed probabilities.

# 5. REFERENCES


[1] J. T. Tou, R. C. Gonzalez, *Pattern Recognition Principles*, Reading MA: Addison-Wesley, 1974.

[2] M. Srinivas and L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Syst., Man, and Cybernetics*, vol. 24, no. 4, pp. 656-667, Apr. 1994.

[3] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst., Man, and Cybernetics,* vol. 16, no. 1, pp. 122-128, Jan. 1986.

[4] J. Zhang, H. Chung, W. L. Lo, S.Y.R. Hui, and A. Wu, "Implementation of a decoupled optimization technique for design of switching regulators using genetic algorithm," *IEEE Trans. Power Electron*., Nov. 2001 issue.

[5] Jon Reed, "Simulation of Biological Evolution and Machine Learning", *J. Theoret. Biol.,* vol. 17, 319-342, 1967.