Dual Response Approach in Process Capability based on Hybrid Neural Network-Genetic Algorithms

Tritiya A.R. Arungpadang[1], **Stenly Tangkuman**[2] **and Lily S. Patras**[3]

[1,2]Department of Mechanical Engineering, Faculty of Engineering,
Universitas Sam Ratulangi, Manado 95115, INDONESIA

[2]Department of Electrical Engineering, Faculty of Engineering,
Universitas Sam Ratulangi, Manado 95115, INDONESIA

E-mail: [1]tritiya_arungpadang@unsrat.ac.id

Abstract. Process capability has long been recognized as an important performance measure to prove how well the process meets the requirements. Process capability can be improved by applying *dual response approach*, to determine optimal input factors. Using of *artificial intelligence* can optimize the prediction of the best input combination with a limited number of experiments. This study proposes an alternatives procedure using a *dual response approach* and *artificial intelligence*. One of the most common robust design models has been formulated to minimize variability while maintaining the mean on the desired target. A study case was selected to implement the proposed approach and compare it with conventional optimization models to show the improvement in procedures.

1. Introduction

Process capability has long been recognized as an important performance measure to prove how well the process meets the requirements. Process capability can be improved by applying *dual response approach* (DRA) to determine the optimal combination of input factors. *Artificial intelligence* can optimize the prediction of the best input combined with a limited number of experiments.

It has been a regular execution to set a target level of process capability. Some indices have been performed to quantify the capability, and the most frequently used process capability indices (PCI) are C_{pk} and C_{pm}. The setting of input variables maximizing process capability possibly obtained through optimization modeling.

Kwon et.al. [2008] have proposed optimization strategy for robust design based on desirability function introduced by Harrington [1965],

$$\begin{aligned} &\text{Min } \ (d_\mu x d_\sigma)^{1/2} \\ &\text{s.t.} \quad \mu_{min} < \ \hat{\mu}(x) < \mu_{max} \\ &\sigma_{min} < \ \hat{\sigma}(x) < \sigma_{max} \\ &\text{x} \in \ \Omega \end{aligned} \qquad \dots (1)$$

where the desirability functions for the process mean and standard deviation (SD) are indicated by d_μ, and d_σ, .

However, it is often that decision makers may confront with difficulties in assigning the weights and coefficients related with preference of individual objectives [Kim and Cho, 2005].

Arungpadang and Kim [2012] have used robust parameter design and back propagation neural network (BPNN) based on historical data. Some limitations can be found in the step of BPNN training, where BPNN is often trapped in the minimum local due to the use of gradient descent [Ma and Su, 2010]. Combination research between neural network and genetic algorithm began in around 1988s. The use of GA aims to improve the performance of NNs through determining the optimal NN architecture or setting parameters as an alternative to optimizing networks [Sexton et.al., 2004]. This combination (NN-GA) is very well used for integrated process modeling and optimization, especially for complex process models. This study proposes the use of capability indices as an optimization metric for the purpose of process design optimization.

2. Hybrid BPNN-GA Procedure

The DRA based hybrid NN-GA using in this study consists of three stages, firstly, happenstance (or experimental) data are collected to construct a BPNN to represent the input-response relationship, based on the predicted mean and SD responses, for a given setting of input factors within the feasible solution space. Secondly, the population initialization is conducted. Afterwards, the fitness computation and scaling are carried out. Then GA's procedures are applied. Finally, the optimum parameter set are obtained based on the mean and SD response. The overall procedure of the proposed approach is described in Figure 1.

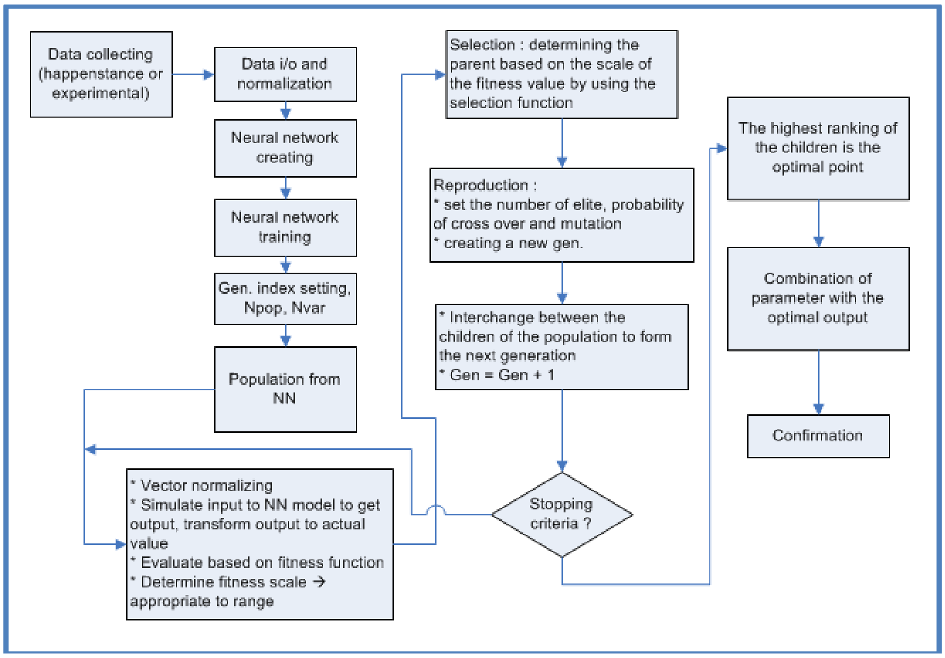

Data collecting (happenstance or experimental)
Data i/o and normalization
Neural network creating
Neural network training
Gen. index setting, Npop, Nvar
Population from NN
* Vector normalizing
* Simulate input to NN model to get output, transform output to actual value
* Evaluate based on fitness function
* Determine fitness scale → appropriate to range
Selection : determining the parent based on the scale of the fitness value by using the selection function
Reproduction :
* set the number of elite, probability of cross over and mutation
* creating a new gen.
* Interchange between the children of the population to form the next generation
* Gen = Gen + 1
Stopping criteria ?
The highest ranking of the children is the optimal point
Combination of parameter with the optimal output
Confirmation

Figure 1: Hybrid BPNN-GA approach

2.1. Construct and Training BPNN

Determine experimental data used for training and testing the BPNN model. Establish the network architecture (input, hidden, and output layers). Search the optimal network architecture and confirm that the network is not overfitted. Convert the experimental input and output data to number within the range [-1, 1]. The normalization is performed in order to avoid mismatch between the influence of some input values and the network weights and biases. Then, train the network using the normalized data by utilizing a training function of Levenberg-Marquardt.

2.2. Population Initializing, Fitness Computation and Scaling

Adjust the initial generation index to zero, the populatian number and the number of independent variables. Generate a random initial population. Every individual has vector entries with certain lengths which are divided into many segments. Next, the performance of the solution vector in the current population is computed by using a fitness function. Convert the solution vector to number [-1,1]. Then, the vector is entered as an input vector to the training process to obtain the corresponding outputs. After that, scale the raw fitness scores to values in a range that is appropriate for the selection function. In the GA, the selection function applies the scaled fitness values to pick the parents for the next generation. The performance of the GA is affected by the range of the scaled values. The scaling function employed in this algorithm is based on the rank of each individual by its score. Lower raw scores have higher scaled values since the algorithm minimizes the fitness function.

2.3. Genetic Algorithms Procedures

Select the parents based on their scaled values by taking the selection function. The selection function specifies a higher probability of selection to individuals with higher scaled values. Each individual can be chosen more than once as a parent.

Options of reproduction specify how the GA produces children for the next generation from the parents. Elite count specifies the number of individuals with the best fitness values that are assured to withstand to the next generation. Arrange elite count to be a positive integer within the range (elite children). Cross over fraction decides the fraction of each population that are produced by crossover. The rest of individuals in the next generation are produced by mutation. Set crossover fraction to be a fraction between 0 and 1. Cross over allows the algorithm to extract the best genes from different individuals. That process occurs by choosing genes from a pair of individuals in the current generation and recombining them. The output is potentially superior children for the next generation. The probability is equal to cross over fraction. Mutation function performs small random changes. It gives genetic diversity and therewith increases the possibility to create individuals with better fitness values.

The recent population is replaced with the children to form the next generation since the reproduction is made. Next process is the increment of the generation index (Gen = Gen + 1). Then, repeat the fitness computation stage to increment of generation stage, until convergence is achieved. The algorithm stops if one of the following five conditions is met. They are fixed generations, fitness limit, times limit, stall generations or stall time limit. If the convergence criterion is achieved, the children with the highest ranking based on the fitness value are decided to be the optimal parameter set of the population.

3. Process Capability Procedure

Based on procedures have explained before, optimum setting can be obtained. The optimum parameter set can be processed using optimization metric to maximize the process capability of the least capable process. The procedure of the proposed approach is described in Figure 2.

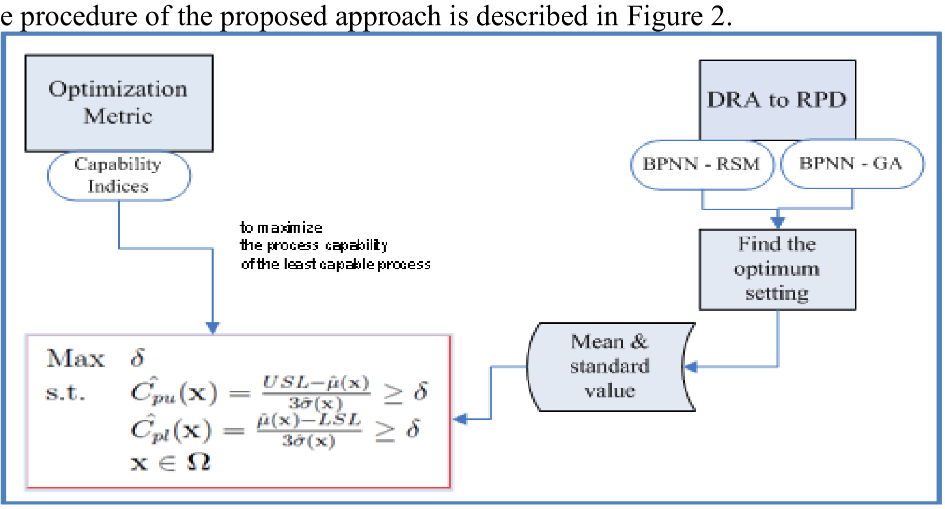

Optimization Metric
Capability Indices
to maximize the process capability of the least capable process
DRA to RPD
BPNN - RSM
BPNN - GA
Find the optimum setting
Mean & standard value
Max δ
s.t. $\hat{C_{pu}}(\mathbf{x}) = \frac{USL-\hat{\mu}(\mathbf{x})}{3\hat{\sigma}(\mathbf{x})} \geq \delta$
$\hat{C_{pl}}(\mathbf{x}) = \frac{\hat{\mu}(\mathbf{x})-LSL}{3\hat{\sigma}(\mathbf{x})} \geq \delta$
$\mathbf{x} \in \Omega$

Figure 2: BPNN-GA to Process Capability Procedure

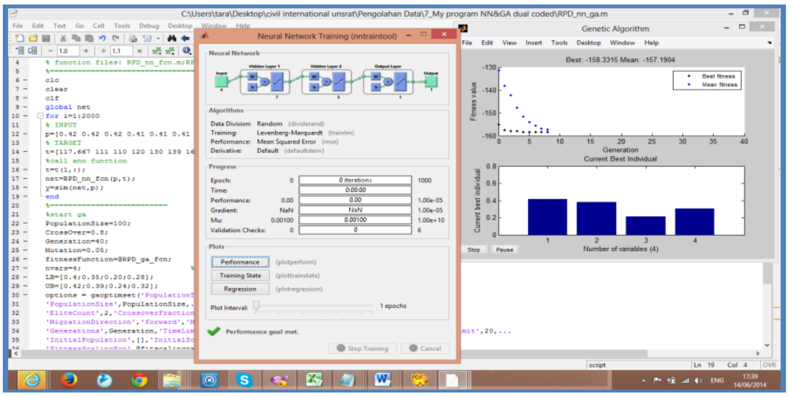
Process capability index is determined by [Plante, 2001],

$$C_{pk}^{(i)} = \min\left\{C_{pu}^{(i)}, C_{pl}^{(i)}\right\} = \min\left\{\frac{USL_i-\mu_i}{3\sigma_i}, \frac{\mu_i-LSL_i}{3\sigma_i}\right\} \qquad \ldots (2)$$

4. Illustrative Example

To perform the proposed procedure in Figure 1 and Figure 2, a case study in Arungpadang and Kim (2013a) and Arungpadang and Kim (2013b) are followed-up. A pharmaceutical process in Subramanian et al. (2004), discussed about *cytarabine liposomes*, had three input variables. They are ***drug/lipid molar ratio*** (X_1), ***PC/Chol in percentage ratio of total lipids*** (X_2), and the ***volume of hydration medium*** (X_3). The output variable of interest is the ***percentage drug entrapment*** (PDE) of which mean and standard error of mean (SEM) are also given.

Global Optimization Toolbox of Matlab 2012b provides methods to search for global solutions for problems that contain multiple maxima or minima, which includes the GA solvers. This solver supports algorithmic customization. A custom GA variant can be created by modifying initial population and fitness scaling options or by defining parent selection, crossover, and mutation functions. In the next step, hybrid BPNN-GA model is simulated to obtain the optimal parameter within the feasible solution space of the system. The values of the three input variables are set as continuous and fall in the range [-1,1]. The operational conditions of GA are set as: population size, cross over fraction, number of generation, fitness scaling function, selection function, crossover function, mutation function and mutation probability. The display of coding in m-file and the running result windows from hybrid BPNN-GA model can be seen in Figure 3.



C:\Users\tara\Desktop\civil international unsrat\Pengolahan Data\7_My program NN&GA dual coded\RPD_nn_ga.m

```
% function files: RPD_nn_fcn.m;RP
%-------------------------------
clc
clear
clf
global net
for i=1:2000
% INPUT
p=[0.42 0.42 0.42 0.41 0.41 0.41
% TARGET
t=[117.667 111 110 120 130 139 16
%call ann function
t=t(1,:);
net=RPD_nn_fcn(p,t);
y=sim(net,p);
end
%-----------------------------
%start ga
PopulationSize=100;
CrossOver=0.8;
Generation=40;
Mutation=0.05;
fitnessFunction=@RPD_ga_fcn;
nvars=4;
LB=[0.4;0.35;0.20;0.28];
UB=[0.42;0.39;0.24;0.32];
options = gaoptimset('PopulationT
'PopulationSize',PopulationSize,.
'EliteCount',2,'CrossoverFraction
'MigrationDirection','forward','M
'Generations',Generation,'TimeLim                mit',20,...
'InitialPopulation',[],'InitialSc
```

Neural Network Training (nntraintool)

Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress

Epoch:	0	0 iterations	1000
Time:		0:00:00	
Performance:	0.00	0.00	1.00e-05
Gradient:	NaN	NaN	1.00e-05
Mu:	0.00100	0.00100	1.00e+10
Validation Checks:	0	0	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs

✓ Performance goal met.

Genetic Algorithm

Best: -158.3315 Mean: -157.1904

Figure 3: Matlab's m-file and result

5. Discussion

Referring to the analysis, hybrid BPNN-GA model can predict quite well the mean and SD value. The result needs to be validated by conducting a confirmation experiment to see if the desired responses are obtained. The most important thing is hybrid BPNN-GA model gives an alternative method to solve and predict the optimal parameters despite limited number of experimental runs, which allows decision making to be taken quickly, reduces the total cost and shorten the process time.

Using the equation (2) as an objective function, an optimization model for process capability is written as,

$$\begin{aligned} &\text{Max} \quad \delta \\ &\text{s.t.} \quad \hat{C}_{pu}(x) = \frac{USL-\hat{\mu}(x)}{3\hat{\sigma}(x)} \geq \delta \end{aligned} \qquad \ldots (3)$$

$$\hat{C}_{pl}(x) = \frac{\hat{\mu}(x) - LSL}{3\hat{\sigma}(x)} \geq \delta$$
$$x \in \Omega$$

Eq. (3) can be more appealing in practical applications because this capability indices are widely applied as a key performance measure. Using this equation, no need any information regarding importance preference. Most companies designate a desirable level of process capability.

Assume that at least 77.0% of PDE is required for this process, the output variable is the *larger-the-better* characteristic and the lower specification limit is given by 77.0. The optimal solution to the optimization model in equation (3) is (X_1, X_2, X_3) = (1:12.41; 60.8:39.2; 1.75ml) with the mean of 84.58 and the SD of 1.79, which induces the process capability (C_{pk}) of 1.41. For desirability model in eq. (1) requiring more information from decision maker to set up the optimization model. Assume that user requirements, r = s = t = 1.0, μ_{min} = 77.0, τ = 85.0, σ_{min} = 1.0, and σ_{max} = 2.5. The optimal solution to the model in eq. (1) is (X_1, X_2, X_3) = (1:13; 60:40; 2ml) with the mean of 81.92 and the SD of 1.43, of which process capability is only 1.14. Although the SD of the PCI-based model is larger than the desirability-based model, a more capable process can be realized without more information from the decision maker.

6. Conclusion

A DRA to robust design have investigated using the process capability as an optimization metric to produce a compromise solution. It has been shown that the PCI-based robust design provides a more capable process. Furthermore, the model does not need more information considering decision makers' input. This approach may be useful in many practical applications where the process capability is an important criterion in process or product design.

Acknowledgements

This research is supported by Sam Ratulangi University and Indonesian Ministry of Research, Technology and Higher Education through DIPA UNSRAT funding number SP DIPA-042.01.2.400959/2018.

References

Arungpadang, T.R., & Kim, Y.J. (2012), *Robust Parameter Design Based on Back Propagation Neural Network*, Management Science (The Korean OR/MS Society)*, **29**(3), pp. 81-89.*

Arungpadang, T.R., & Kim, Y.J. (2013a), *A Study on Dual Response Approach Combining Neural Network and Genetic Algorithms*, Journal of the Korean Institute of Industrial Engineers*, 39, pp. 361-366.*

Arungpadang, T.R., & Kim, Y.J. (2013b) *Design Optimization for Capability Enhancement based on a Dual Response Approach*, ICIC Express Letters*, 7(5), pp. 1731-1736.*

Harrington, E.C. (1965) *The Desirability Function*, Industrial Quality Control*, 21, pp. 494-498.*

Kim, Y.J., & Cho, B.R. (2005) *Trade-off Strategies in Robust Design Optimization Problems*, Journal of the Korean Maintenance Management Society*, 10, pp. 69-74.*

Kwon, Y.J., Kim, Y.J., & Cha, M.S. (2008) *Desirability Function Modeling for Dual Response Surface Approach to Robust Design*, Industrial Engineering and Management Systems*, 7, pp. 197-203.*

Ma, H.Y., & Su, C.T. (2010) *Applying Hierarchical Genetic Algorithm Based Neural Network and Multiple Objective Evolutionary Algorithm to Optimize Parameter Design with Dynamic Characteristics*, Journal of Quality, *17 (4), pp. 311-325.*

Plante, R.D., (2001) *Process Capability: a Criterion for Optimizing Multiple Response Product and Process Design*, IIE Transactions*, 33, pp. 497-509.*

Sexton, R.S., Dorsey, R.E., & Sikander, N.A. (2004) *Simultaneous Optimization of Neural Network Function and Architecture Algorithm*, Decision Support Systems*, 36 (3), pp. 283-296.*

Subramanian, N., Yajnik, A., & Murthy, R.S. (2004) *Artificial Neural Network as an Alternative to Multiple Regression Analysis in Optimizing Formulation Parameters of Cytarabine Liposomes*, AAPS PharmSci Tech*,* ***5****(1), E4.*

Vining, G.G., & Myers, R.H. (1990) *Combining Taguchi and Response Surface Philosophies : A Dual Response Approach*, Journal of Quality Technology*,* ***22****(1), pp. 38-45.*