

ANDRZEJ SKOWRONEK*

EVALUATING THE INFLUENCE OF GENERATION METHODS ON THE QUALITY OF THE SMART DESIGNS OF EXPERIMENT

OCENA WPŁYWU METOD GENEROWANIA NA JAKOŚĆ ELASTYCZNYCH PLANÓW EKSPERYMENTU

Abstract

This paper presents results of computer simulation made to evaluate how the quality of the smart designs of experiment depends on the methods of generating. The quality of smart designs was evaluated by comparing the known values of special testing functions simulating the real research object and approximated values predicted by neural networks trained with the sets based on smart designs of experiment. The results suggest the possibility of significant reducing the number of experiments runs using studied smart designs.

Keywords: smart design of experiment, experimental research, neural approximation

Streszczenie

Artykuł przedstawia wyniki symulacji komputerowej, której celem było dokonanie oceny wpływu metody generowania elastycznych planów eksperymentu na ich jakość. Jakość planów oceniana była na podstawie porównania znanych wartości funkcji testowych symulujących rzeczywisty obiekt badań i wartości aproksymowanych zwracanych przez sieć neuronową uczoną z wykorzystaniem zbiorów danych opartych na analizowanych planach. Wyniki sugerują możliwość znacznej redukcji rozmiaru potencjalnych badań eksperymentalnych planowanych z wykorzystaniem analizowanych planów elastycznych.

Słowa kluczowe: elastyczny plan eksperymentu, badania eksperymentalne, aproksymacja neuronowa

* PhD. Eng. Andrzej Skowronek, Institute of Applied Informatics, Faculty of Mechanical Engineering, Cracow University of Technology.

1. Introduction

Experimental research play in many areas of science a key role in obtaining the knowledge and information. Using techniques offered by theory of experiment you can support conducting experimental research, especially when it is necessary to limit the time of experiment or cost. In the theory of experiment there are used various types of experimental designs. The type of design depends largely on the specific purpose of research ([1, 2, 3]), which could be for instance looking for unknown object's function, verification how inputs effect output and looking for extremes of object's function. Application the experimental designs with special techniques for analysis of experiment's results can often facilitate reducing the size of the experiment (number of runs, observations, etc.) and obtaining relevant information from research, without reducing their quality ([4, 5]). In addition to the undoubted benefits of using traditional designs of experiment (reducing the number of experiment's runs, reducing time and cost of research), you can also see some negative effects in case of conducting experiment according to them. The researcher can not change number of design's units, number of inputs' levels and has to carry out the experiment strictly according to used design. A quite different approach to the concept of planning the experiment enable smart designs of experiments ([6]) which allow the researcher to set the number of design's units and number of inputs' levels.

2. Smart designs of experiment

Smart designs of experiment are generated in a dedicated computer application. Creating of smart design consists of the following steps [7]:

- defining characteristics of the design: number of inputs (factors), number of designs units, number of inputs levels
- generating of inputs' levels according to chosen method
- generating of sets of inputs factors levels
- generating of set of all possible designs' units by permuting all inputs levels
- completing the design by selecting from the set of all possible design's units only fulfilling the special condition
- equipartitional analysis to evaluate quality of design (quality means regular and equipartitional distribution of design's units in inputs space).

Smart designs are generated basing on three important principles: adaptation, randomness and equipartition ([6, 7]). The first principle means the possibility of adjusting the design's characteristics to the conditions of the experiment and characteristics of the analyzed object, what was discussed above. The second principle means that smart designs are created in non-deterministic way: both generation of inputs levels and selection of designs units are conducted using pseudo-random numbers. However, there are some limitations put on the random way of generation of design's units:

- using a parameter called "important difference" (Δx), a minimal permissible distance between currently generated value and existing values of each input factor levels (Fig. 1),
- a parameter called "minimal Euclid's distance" (*esmin*) – it is Euclid's distance to the nearest "neighbour-unit" in the inputs space, calculated for each design's unit, each unit must fulfill the condition: $es \geq esmin$ (Fig. 2).

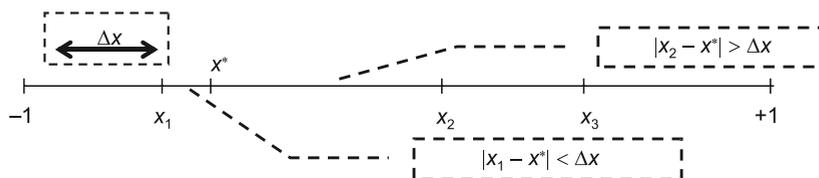


Fig. 1. Value x^* doesn't pass the important difference condition test and will be removed

Rys. 1. Wartość x^* nie spełnia kryterium różnicy istotnej i zostanie usunięta

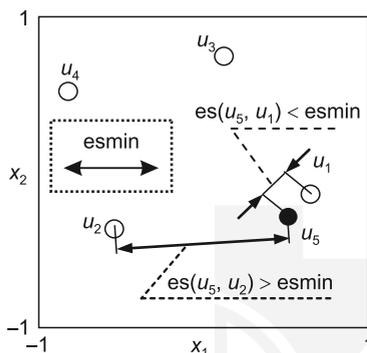


Fig. 2. Unit u_5 fails the $es \geq esmin$ condition test $es(u_5, u_1) < esmin$ and will be removed

Rys. 2. Układ u_5 nie spełnia warunku $es \geq esmin$ i zostanie usunięty

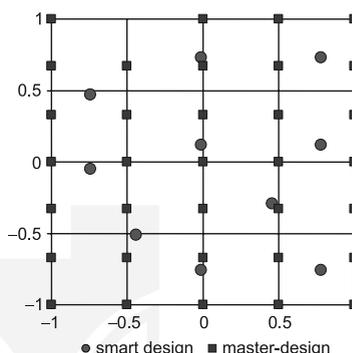


Fig. 3. 2-inputs master-design and smart design

Rys. 3. Plan wzorcowy o 2 wielkościach wejściowych i plan elastyczny

The conceptions of both described above parameters are based on a conception of Euclid's distance and they use the fact that a set of experimental design units in the space of inputs is equivalent to the set of points in orthogonal coordinate system and the combinations of inputs' levels (which make up units of designs) are equivalent to points coordinates. The Δx and $esmin$ parameters support equipartition of the designs units in the inputs space. If there are no other assumptions designs units should cover regularly the whole inputs space (the third rule). To estimate the regularity of the distribution of the designs units it is used the method of the equipartitional analysis (EPA, [6]). The analysed (created) experimental design is compared with the master-design which units are distributed perfectly regularly in the inputs space ([6, 7], Fig. 3). The master-designs have the same numbers of inputs as the analysed designs, the same number of various inputs levels but the number of design's units is often significant higher and equal to the product of numbers of all inputs levels. However, the levels of master-design are calculated for each input by dividing the length of input range by the number of input levels (Fig. 3). For each unit of the master-design you can evaluate the Euclid's distance to the nearest unit of the analysed design. Next you can evaluate for such a collection (called equipartitional set) a lot of statistical parameters (for example descriptive statistics) or make one of the statistical tests (for example goodness of fit test, see: [8]). Each

of them could be an equipartition criterion in this analysis. In this paper there were used two parameters: maximal (*elmax*) and mean (*elmean*) value of equipartitional set. The *elmean* parameter describes the central tendency of equipartitional set and the *elmax* parameter gives information if there are some huge empty areas in the input space (without designs units), what is important taking into consideration the assumption that the designs units should cover the whole inputs space. The dependence between the both equipartitional analysis parameters and the designs quality (quality means equipartition, perfect regularity of the designs units in the inputs space) was verified and described in [7]. The conclusion was: the less value of equipartitional parameter the more regular distribution of design's units in inputs space.

There are actually three ways to generate the inputs' levels ([9]). In the first method ("Z" method) inputs levels are generated as pseudo-random values from the normalized range $[-1, 1]$ and checked if they pass the important difference condition. If a value fail the test, it is removed and the next one is generated to reach the right amount. In "R" method levels of inputs are calculated by dividing the inputs ranges by the demanded numbers of inputs levels. The smallest level is calculated as the minimum of input's range and the biggest (last) level is calculated as maximum of input's range. In the R2-method the idea of levels calculating is that each level should be the center point of equal areas of influence. The first and the last levels are not equal to minimum and maximum of input range (Tab.1).

Table 1

Examples of levels in R and R2 methods

Method	Values used in case of 5 levels				
R	-1	-0.5	0	0.5	1
R2	-0.8	-0.4	0	0,4	0.8

The smart designs generator in the current version has implemented functionalities that support selecting the optimal values of important generation's parameters – important difference (Δx , used in Z-method of levels generating) and minimal Euclid's distance (*esmin*, used to ensure high regularity and equipartition of design's units in the inputs' space). In the previous versions of generator researcher must set it by himself. If he doesn't know well the principles of functioning the smart designs generator and doesn't have some intuition or experience in designs generating process it is likely that the generated design won't be optimal – designs units won't cover equally the whole inputs space. In a case of setting to small values of generation parameters equipartition decreases or it is not possible to get experimental design with the assumed number of units otherwise.

If inputs levels are generated with Z-method, the initial value of Δx is calculated as:

$$\Delta x = \frac{r}{n_k} \quad (1)$$

where:

- r – a length of inputs range $[-1..1]$,
- n_k – a number of levels for each of $1..k$ inputs.

If it is not possible to generate all the levels, the initial value is reduced by 10% and the process of levels generating starts again to obtain the demanded number of inputs levels.

The *esmin* parameter is calculated similar to Δx parameter. The initial value is calculated according to the formula:

$$esmin = \sqrt{\sum_{j=k}^i \frac{r}{k_j}} \cdot \left(2 - \frac{u}{\prod_{j=1}^i k_j} \right) \quad (2)$$

where:

- r – means length of inputs range, which is usually normalized to $[-1, 1]$,
- k_j – means number of levels for j -input,
- u – means number of designs units,
- i – means number of inputs,
- $j = 1..i$.

The next step is generating series of designs fulfilling the *esmin* condition. If at least one design can be created, *esmin* value is automatically increased and new designs are generated again. If any design is created, *esmin* value is automatically decreased and designs are generated again. In the case of increasing *esmin* value, if the new generated design has a better quality than the previous, *esmin* value is increased again and new designs are generated once again. If the new generated design has worse quality than the previous, new design is not generated again and the previous (with better quality) design is saved. In the case of decreasing *esmin* value, if at least one new design can be generated, new designs are not generated again. If any new design can be generated, *esmin* value is automatically decreased again and new designs are generated once again.

The better quality design is selected basing on equipartitional parameters *eImax* and *eImean*. The lower values of these parameters means better design's quality (more regular distribution of unit, equally and without excessive concentration in the inputs' space). However, it is a question which of to applied parameters is more important. The answer is not easy and could depend on the researcher's preferences or conditions of the conducted experiment. In this research the design is selected as having the better quality if one of two conditions selected by the generator's user is true:

- either both of parameters *eImax* and *eImean* have the lower values than the best previous,
- or *eImax* parameter is lower than 95% the best previous and *eIsr* parameter is lower than 105% the previous best.

The second condition could be applied if the generator's user prefers especially *eImax* parameter, which is calculated as a maximal value of equipartitional set and can be interpreted as an indicator of big areas in the input's space without any design's unit.

The smart designs of experiment are multiple-generated ([7]). The reason is application of pseudo-random numbers in algorithm of designs generating. Designs generated with the same seed of pseudo-random number generator, the same parameters of generation (Δx , method of inputs levels generating, *esmin*) and the same design's characteristic (number of inputs, number of inputs levels, number of design's units) will be identical. But if you change the seed value or just if you try to generate it next time even with the same generation parameters,

they could be different and the difference of design's quality could be sometimes significant. To avoid such a problem it seems to be necessary to generate several designs and choose one basing on EPA-parameters (*e1max* and *e1mean*). That is the idea of multi-generated smart designs of experiment. The researcher can set pseudo-random number generator seed value by oneself or can generate it basing on real-time clock. To generate identical design again you need only know its seed value. The researcher can select the EPA-parameter which he prefers to choose the best design. Each design generating is repeated up to 20 times to get 10 designs.

3. Computer simulation

To evaluate quality of the smart designs of experiment it was carried out a computer simulation. In the simulation instead of using the real object of research, the special test functions were used. It was generated a series of smart designs with 2 inputs, 15 units (cases) and inputs' levels calculated according to Z, R and R2 methods. In the simulation two testing functions were used to simulate the real research object: SumOfSquare function (first function of De Jong's, Eq. 3) and Rosenbrock's function (2. function of De Jong's, Eq. 4) [10].

$$f(x1, x2) = (1 - x1)^2 + 100 \cdot (x2 - x1^2)^2 \quad (3)$$

$$f(x1, x2) = x1^2 + x2^2 \quad (4)$$

The real values of testing functions were compared to approximated values. Approximation was conducted with application of neural networks. The learning sets were built basing on values of testing functions calculated for each smart design unit. Additionally as a base of learning set was used 2-inputs 5-levels design (called "K"), generated according to R-method but with 25 units, created as all possible unique combinations of both inputs levels. The approximated values were calculated by the trained neural networks as predicted for a special 121-elements testing set (all unique combinations of values $\{-1, -0.9, \dots, 1\}$). The differences between real and approximated values were saved as approximation errors sets. Basing on error sets various statistics were calculated: maximal value of errors set, average value, standard deviation of errors set, correlation coefficient between the testing function values and approximated values.

4. Results of simulation

Results of simulation for all generation methods (Z, R, R2 for 15 units and K for 25 units) show Tab. 2 (for SumOfSquare function) and Tab. 3 (for Rosenbrock's function). Both testing functions were normalized to the range $[0, 1]$, so for example the value 0.37 of the maximal error obtained in case of method "Z" of inputs levels generation means 37% difference between the real and approximated values for one of testing cases. Analyzing the maximal errors you can notice big differences between values obtained for three methods used to generate inputs' levels. The best results for 15-units designs were observed for "R" method and it is confirmed by the other statistics. As you can see, for the Rosenbrock's function there are generally generated larger errors than for SumOfSquare function. It is

important, that the errors statistics for “R” method are not significantly worse than those obtained for “K” method, despite the size of training set was 15 units instead of 25, what means a serious reduction of runs in conducted experiment and possibly reduction in cost of experiment.

Table 2

Statistics calculated for approximation error sets for SumOfSquare function				
	Z	R	R2	K
maximal error	0,37	0,13	0,50	0,15
average error	0,06	0,03	0,12	0,02

Table 3

Statistics calculated for approximation error sets for Rosenbrock’s function				
	Z	R	R2	K
maximal error	0,50	0,24	0,30	0,24
average error	0,10	0,07	0,05	0,03

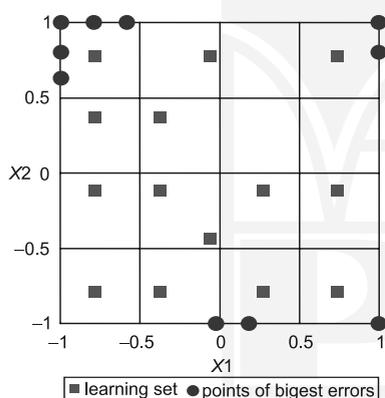


Fig. 4. Places of 10 biggest errors and learning set for SumOfSquare function

Rys 4. Miejsca położenia 10 największych błędów i zbiór uczący dla funkcji SumOfSquare

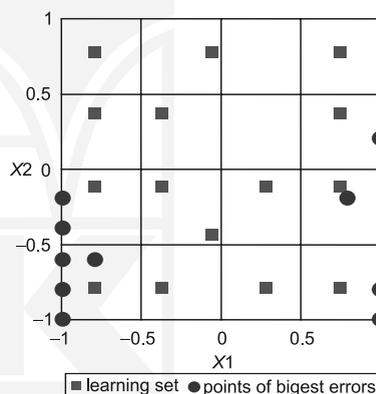


Fig. 5. Places of 10 biggest errors and learning set for Rosenbrock’s function

Rys. 5. Miejsca położenia 10 największych błędów i zbiór uczący dla funkcji Rosenbrocka

Fig. 4 and 5 show learning set and the places of the 10 biggest errors – differences between real and approximated values evaluated for the testing set for “Z” method and both testing function. The conclusions after analyzing points of 10 biggest errors are that they are noticed in the edges of inputs range. Two reasons of the fact are possible. The first reason could be a lack of close “learning data points” what could have an influence on approximated values errors. The second cause could be strong increasing or decreasing of testing function’s values in the area of the significant errors.

5. Conclusions

Among the analyzed three methods of generating the inputs levels especially “R” method is high recommended because the obtained approximation errors were not significantly worse than obtained for “K” method. It is important especially if we want to limit cost or time of experiment by reducing the number of experiment’s runs. The difference in number of design’s units which were bases of training set in neural approximation in the simulation was 40 percent. Advantageous feature of “R” method seems to be that the designs units are distributed very regular in inputs space and some of them are placed on the edges of inputs’ ranges, what could be important if the research object’s function is a varied and unpredictable shape in that area. The simulation was conducted for smart designs of experiment with two inputs, five levels of each input and 15 units. However, the procedures of generating and analysis smart designs are universal for all designs’ characteristics, so the conclusions are true for cases of another number of inputs, levels or design’s units and can be used in a broad area of engineering if you need to conduct an experimental research. The quality of smart design’s of experiment was evaluated basing on neural approximation errors. The neural networks were created and learn according to the same default procedures and it was assumed that it does not have a significant influence on the results.

References

- [1] Polański Z., *Design of experiment*, PWN, Warsaw 1984.
- [2] Montgomery C., *Design and Analysis of Experiments*, 8th edition, John Wiley & Sons, 2012.
- [3] Hinkelmann K., Kempthorne O., *Design and Analysis of Experiments, Volume 1, Introduction to Experimental Design*, 2nd Edition, John Wiley and Sons, 2008.
- [4] Hinkelmann K., Kempthorne O., *Design and Analysis of Experiments, Volume 2, Advanced Experimental Design*, John Wiley and Sons, 2005.
- [5] Goos P., Jones B., *Optimal Design of Experiments: A Case Study Approach*, Wiley 2011.
- [6] Polański Z., *Empirical research – methodology and computer support* (in Contemporary metrology, editor J. Barzykowski), WNT, Warsaw 2004.
- [7] Skowronek A., *A method of self-adaptation calculating the value of the smart designs of experiment generation’s parameter*, Polish Journal of Environmental Studies, Vol. 18, No. 3B, 2009.
- [8] Johnson R.A., Bhattacharyya G.K., *Statistics: Principles and Methods*, 2th Edition, John Wiley and Sons, 2010.
- [9] Skowronek A., *The smart designs of experiment generated with circular inputs’s ranges*, Conference Proceeding of Engineering & Education 2009, Białka Tatrzńska, 4–6 November 2009, Problems Of Modern Techniques In Engineering And Education, 2009.
- [10] Pohlheim H., *Examples of objective functions* (http://www.geatbx.com/download/GEATbx_ObjFunExpl_v38.pdf – 25.02.13).