The Solution Space Reduction Using Frequency Domain of Simulated Annealing

Watha Minsan and Pornthep Anussornnitisarn*
Department of Industrial Engineering
Faculty of Engineering Kasetsart University, Thailand
*E-Mail: fengpta@ku.ac.th

Abstract: The objective of this research is used frequency domain for factor screening while simulated annealing method searches for an optimum. Each iteration of simulated annealing, we consider as two problems, primal and shadow problem. Primal problem runs the conventional simulated annealing. On the other hand, a model using input variables that oscillate at different frequencies during a run which called "Shadow Problem". It is indicated by the frequency spectrum of the output variable where run lenght (denoted by) is large enough. For shadow problem, If the output variable is sensitive to changes in a particular input variable, then oscillating of this input variable induces oscillations in the output variable. The after factor screening, the only remaining important input variables will run continuously on the simulated annealing for solving the primal problem. However, the unimportant input variables are assigned to be constant values from the best values of the simulated annealing of the primal problem. We found that the frequency domain of simulated annealing requires quite fewer iterations than conventional simulated annealing. We describe the method, illustrate a nonlinear problem effectiveness at identifying important main effects, two-way interaction, and quadratic term of known model.

Key words: Simulation Optimization, Frequency Domain Experiment, Factor Screening

INTRODUCTION

The Simulated Annealing (SA) is designed to enhance the likelihood of avoiding local minima en route to global minimum of problem. This technique uses magnitudes of random perturbations as reduced-annealed in a controlled manner. Wherever the injected randomness helps prevent convergence to a local minimum by providing a greater "jumpiness" to the algorithm[1]. However, the interpretation of large amounts of input factors can become an intimidating task.

Factor screening is useful when there are many input factors that affect potentially a simulation output or an output variable. However, we believes that relatively less factors and/or interactions provide an adequate function of the system's behavior. An experiment run the model several times. For example, 2^k (or more) prospective configurations as called 2^k factorial design, where k is amount of input factors in problem function.

For quantitative factors, response surface metamodel is applied and analyzed by regression techniques. The techniques just described all have a run-oriented approach. Run-oriented approaches work well for few input-factors problem. For example, a full factorial experiment involving 5 factors requires 32 runs[2]. As we interest in 20 input-factors, then a factorial experiment involving all these factors would requires 1,048,576 runs. For more precise experiment, 3^k factorial is used, it requires over 3,486,784,401 runs. Unfortunately, response surface methodology requires a large number of simulation runs and is supported by very restrictive assumptions on system's behavior[3]. It is not a good technique for factor screening before run the simulated annealing to optimize a large problem as we interest.

An alternative simulation factor screening method, Schruben and Cogliano (1987) introduced Frequency Domain Methodology for screening many factors only few computer runs[4]. A frequency domain method for factor screening is a simulation model. It is a run with input factors that are varied during a run according to sinusoidal oscillations. Different frequencies during a run are assigned for each factor. Whenever the simulation response is sensitive to changes in a particular factor, then oscillating of this factor induces oscillations in the response. The frequency domain experiment permits one to identify an appropriate polynomial model for simulation output. The frequency domain simulation experiments typically will require 2-3 runs for factor screening.

In this paper, we demonstrate how the methodology called Frequency Domain Experimentation (FDE) can screen input factors during the SA run. The proposed technique will reduce a solution space and number iterations of SA.

METHODS

Simulated Annealing Algorithm:

The SA algorithm is based on the analogy between the simulation of the annealing of solids and the problem of solving large combinatorial optimization problems. For this reason the algorithm became known as "Simulated Annealing"[5]. Its algorithm is shown as below

Line 1: Start
Line 2: Select an initial solution X_i;
Line 3: Select an initial temperature $t_0 > 0$;
Line 4: Select a temperature reduction function α;
Line 5: Repeat
Line 6: Repeat
Line 7: Randomly select $X \in N(X_0)$;
Line 8: $\Delta = f(X) - f(X_0)$;
Line 8: If $\Delta < 0$ then $X_0 = X$;
Line 9: Else
Line 10: Generate random r uniformly in range (0, 1);
Line 11: If $r < P_{accept}$ then $X_0 = X$;
Line 12: Endif
Line 13: Endif
Line 14: Until iteration_count = n_repeat;
Line 15: Set $t = \alpha(t)$;
Line 16: Until stopping condition = true
Line 17: End.

Where

X is set of input factors

P_{accept} is probability to accept for energy change

n_repeat is cooling schedule

$\alpha(t)$ is temperature reduction function.

Frequency Domain Experiment Algorithm:

Frequency domain experiment assumes the expected output of a simulation model, which is modeled over an experimental region by a p-order polynomial given by

$$E(Y) = \beta_0 + \beta_1\tau_1 + \beta_2\tau_2 + ... + \beta_q\tau_q, \quad (1)$$

Where

$E(Y)$ is the expected output

τ_j is a term in the p-order polynomial and is a particular product of the nonnegative integer powers of the input factors $X_j, j \in (1, 2, ..., K)$ where the sum of the exponents is not greater than p (e.g., if $p = 5$, $X_1^2 X_2^4$ is not a term)

β is the coefficient of the τ term

q is the number of potential terms; and

X is input factors $X_1, X_2, ..., X_k$.

The equation (1) describes a static relationship between the expected output and configuration of input factors. For frequency domain experiment, this relationship is obtained through static experiments. For each static experiment, the input factors are varied at specific values in the region of each input factors and $E(Y)$ is estimated. This process is not repeated unnecessarily in the experiment.

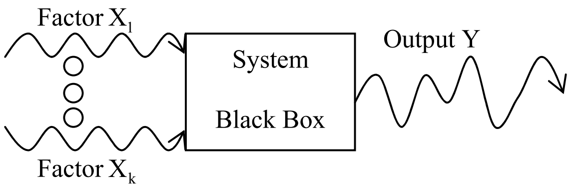

Fig. 1 Frequency Domain Experiment in the Black Box.

Fig. 1, if we view the inputs and outputs of the simulation runs as time series (t) rather than constant values, we can then oscillate the input factors during the simulation run. For each input factor affects the system performance, then the output time series will oscillate at a related driving frequency. Alternatively, for each input factor does not affect performance, then the 'black box' will not transmit the oscillation at a related driving frequency through to the output time series[2].

For the frequency domain factor screening technique, the input factor level setting for each factor $X_j, j \in (1, 2, ..., K)$, (and k is the number of input factors), is varied according to:

$$X_j(t) = X_j(0) + a_j \cos(2\pi\omega_j t). \quad (2)$$

Here,

$t = 0, 1, ..., N-1$, where N is the total number of observations generated by the simulation runs, which is based on time series.

$X_j(0)$ is $0.5(U_j + L_j)$ are the nominal value of factor X_j in time 0, U_j is upper bound of factor X_j and L_j is lower bound of factor X_j

a_j is $0.5(U_j - L_j)$ are the amplitude of the factor X_j

ω_j is the unique driving frequency for factor X_j [6].

It is important to note that equation (2) describes an input factor that oscillates at different frequencies during a run.

After run experiment, we analyze the output $Y(t)$ by using spectral analysis.

Frequency Domain of Simulated Annealing:

We have developed a new technique based on a conventional SA method which is applied a frequency domain experiment for reducing a solution space during the SA run for optimization. This technique is a simulated annealing-based simulation optimization method developed to improve the performance of simulated annealing for continuous variable simulation optimization.

We adapte a simulated annealing algorithm by adding a few commands to collect observations of $Y(t)$ for transforming the time domain into frequency domain and then analyze its spectral frequency. The command are given by

$$X_0^{shadow} = X_0 + a\cos(2\pi\omega_j t), \tag{3}$$

Where

X_0 is set of new input factor values from SA

$a\cos(2\pi\omega_j t)$ is set of term of frequency domain experiment method

And

$$Y^{shadow}(t) = f(X_0^{shadow}), \tag{4}$$

Where

$f(X_0^{shadow})$ is the output response of function assign to $Y^{shadow}(t)$

t is a iteration_count.

Equation (3) and (4) is added to SA algorithm before line 14, respectively. For equation (4), it is called is "Shadow Problem". Whenever we run lengths of simulation run (denoted by n) large enough to include at least 10 full cycles of the lowest term indicator frequency. The larger the value of n, the smaller the variance of the spectral estimators[4]. The run lengths of SA can be made typically large iterations, but experimental cost for run increase very little margin. For example, we run lengths 2^{15} observations of $Y(t)$.

Last added command is a command to analyze spectral frequency of shadow problem. There is given by

If iteration_count = n_check then
$f(\omega) = \text{fft}(Y^{shadow})$;
Check Y'
If X are unimportant inputs then reduce set of X
Endif
Endif

Were

n_check is the maximum run lengths (n)

fft is the fast Fourier transform

$f(\varpi)$ is the spectral term of Y^{shadow}.

The previous command is added after the equation (4). Shadow problem is indicated by the frequency spectrum of $Y^{shadow}(t)$, if the output variable is sensitive to changes in a particular input variable, then oscillating of this input variable induces oscillations in the output variable. The only remaining important input variables will run continuously on the SA and unimportant input variables are assigned to be constant from the best values of the simulated annealing of the primal problem. Then we obtain a smaller set of input variables for solving the SA of the primal problem next.

EXPERIMENT

Suppose that we interest in twenty input factors. A black box system is given as follow

$$\begin{aligned} Y(t) = {} & -4X_1^2 + 25X_2^2 + 50X_3^2 + 100X_4 + 75X_5 \\ & - 50X_6 - 25X_7 + X_8 + X_9 + X_{10} + X_{11} \\ & + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} + X_{17} \\ & + X_{18} - X_{19} - X_{20}. \end{aligned}$$

The following steps summarize the frequency domain of SA procedure

Step 1: Our performance measure is the simulation out put, Y.

Step 2: Twenty input factors will be oscillated during the SA runs to assess their impact on Y.

Step 3: The twenty input factors were oscillated at unique frequencies of $310/2^{15}$, $598/2^{15}$, $763/2^{15}$, $825/2^{15}$, $846/2^{15}$, $949/2^{15}$, $1321/2^{15}$, $2063/2^{15}$, $3054/2^{15}$, $3446/2^{15}$, $4024/2^{15}$, $5303/2^{15}$, $6005/2^{15}$, $6789/2^{15}$, $7882/2^{15}$, $9223/2^{15}$, $11039/2^{15}$, $12546/2^{15}$, $14547/2^{15}$ and $14588/2^{15}$ cycles per time unit, respectively. Results of primary indicator frequencies are the driving frequencies for main effects. Indicator frequencies for quadratic effects are $620/2^{15}$, $1196/2^{15}$, $1526/2^{15}$, $1650/2^{15}$, $1692/2^{15}$, $1898/2^{15}$, $2642/2^{15}$, $4126/2^{15}$, $6108/2^{15}$, $6892/2^{15}$, $8048/2^{15}$, $10606/2^{15}$, $12010/2^{15}$, $13578/2^{15}$, $15764/2^{15}$, $14322/2^{15}$, $10690/2^{15}$, $7676/2^{15}$, $3674/2^{15}$ and $3592/2^{15}$ cycles per time unit, respectively[6].

Step 4: The run length is 2^{15}.

Step 5: We use MATLAB program to compute this experiment.

Figure 2 illustrated the total spectrum of main effect and quadratic effect altogether in each input factors to analyze the whole picture of each factors. We found that input factor X_3 had the highest influence followed by X_1, X_2, X_4, X_5, X_6 and X_7 respectively. These factors had high influence towards the response variable $Y(t)$, when considered with indicated our problem which correlated. This meant that this factor screening method was work well.

RESULTS AND DISCUSSION

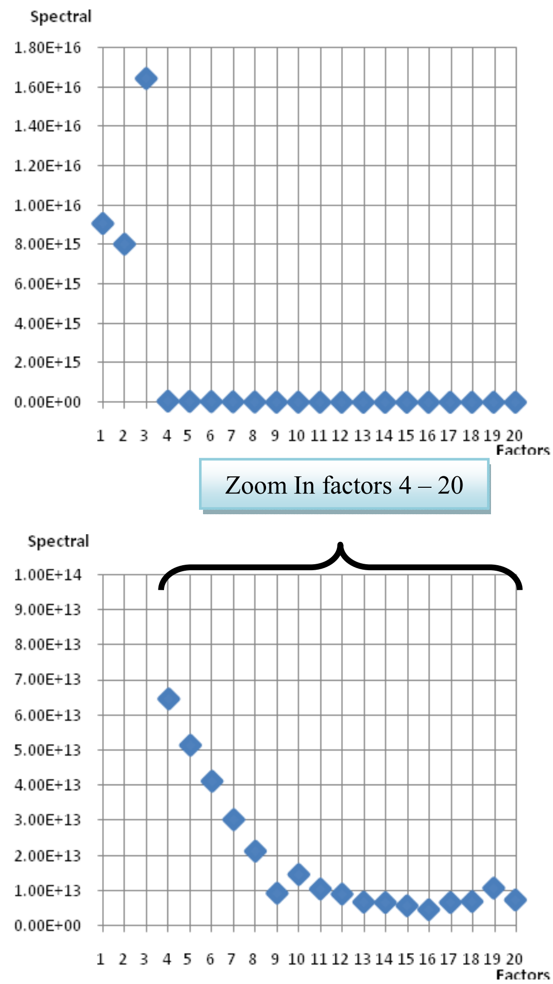

Fig 2. Spectral term of 20 factors with $n = 2^{15}$

We could organize the order of the input factors which had influence towards the response variable $Y(t)$ according to the experiment as follows; $X_3, X_1, X_2, X_4, X_5, X_6, X_7, X_8, X_{10}, X_{19}, X_{11}, X_9, X_{12}$ $X_{20}, X_{18}, X_{13}, X_{17}, X_{14}, X_{15}$, and X_{16} respectively.

From table 1, we determined the reduction of input factors by creating policies. The identified policies were 10 policies altogether. The first policy did not require decreasing in the numbers of input factors. The second policy would like to decrease in the numbers of input factors used in the system by 10%. The third policy to the tenth policy would like to decrease in the number of input factors used in the system by 20% to until 90% respectively. Therefore, creating policies for input factors was convenient to solve a problem in operation management. The fewer input factors would also result in less confusion.

From the result, SA took 8,743,314 iterations to find the solution when compared with frequency domain of SA with each policy; we found that the numbers of iterations spent were fewer. When we determined to find the answer from input factors 90% (the number of input factor 10% were not applied) we received 2 variables that would not be used which were X_{15} and X_{16} respectively (considered from input factors ordering). This policy applied the numbers of iterations in searching the answers 7,554,054 iterations which were decreased about 1 million iterations from the previous one. When we determined to decrease the numbers of input factors continuously, it was found that the numbers of iterations were also decreased. At the 10th policy, the numbers of input factors were at 10% (the number of input factor 90% were not applied), the numbers of iterations spent were only 38,900 rounds.

From the indicated statement problem, although the decrease in number of the variables, the effects to the answer when compare to global optimization there was only a few percentage different. From table 1, you will see that although we reduced the number of input factors down to 10%. The answer was different from global optimization only around 18%. However, the effects towards the answer would vary amongst each problem statements, a researcher should consider carefully the number of input factors to be decreased by considering spectrum graph of frequency domain according to figure 2.

From figure 4, we test running on 4 computers with different CPU applying the same seed. We found that computers with many cores were faster than one core computers. However, no matter types of computer used for input factors of 20 variables would spend so much time. The best computer still took more than 1 hour for computing the answer of this statement problem, but if the numbers of input factors were reduced to 10%, it would take only less than 1 minute.

CONCLUSION

We have shown that heuristic and factor screening could be able to implement at the same time by not taking more time in processing process. Our aim was to encourage researchers and practitioners applied this technique in operation management or other related fields. For example, in an industrial plant, the numbers of relevant factors might be high so it might be difficult to find a suitable set up for the mechanic. The decrease in variables would be benefit and convenient.

However, frequency domain experiment was not widely used as most of researchers and practitioners could not foresee a benefit of changing from time domain to frequency domain which normally would be complicated. We would like to show that frequency domain was not that complicated and at present there were a lot of equipment to help transform to frequency domain. In this research, we used MATLAB to transform with function Fast

Fourier Transform (fft) which the numbers of data 2^{15} took only 2 seconds.

Our next research will be an experiment on factor screening by other methods such as Genetic Algorithm or Tabu Search considering discrete event input factors in the problem statement as well.

Table 1 Optimization results: identification of policy

Policy	Minimum Objective value	Difference from Global Optimum solution		Number of Iterations
		% Diff.[1]	Value Diff.[2]	
Simulated Annealing	-4,000,262,999,390.78	-0.0000000152	609.22	8,743,314.00
Policy 1 (not reduced)	-4,000,262,999,390.78	-0.0000000152	609.22	8,005,826.00
Policy 2 (18 Vars.)	-4,000,261,102,582.43	-0.0000474323	1,897,417.57	7,544,054.00
Policy 3 (16 Vars.)	-4,000,258,728,903.01	-0.0001067704	4,271,096.99	5,287,136.00
Policy 4 (14 Vars.)	-4,000,256,686,258.68	-0.0001578332	6,313,741.32	5,271,654.00
Policy 5 (12 Vars.)	-4,000,254,577,112.57	-0.0002105583	8,422,887.43	5,191,734.00
Policy 6 (10 Vars.)	-4,000,252,253,531.09	-0.0002686441	10,746,468.91	3,229,100.00
Policy 7 (8 Vars.)	-4,000,250,438,669.06	-0.0003140126	12,561,330.94	1,979,650.00
Policy 8 (6 Vars.)	-4,000,239,940,167.53	-0.0005764579	23,059,832.47	3,534,780.00
Policy 9 (4 Vars.)	-4,000,091,111,067.69	-0.0042969408	171,888,932.31	1,034,363.00
Policy 10 (2 Vars.)	-3,305,577,930,663.05	-17.3659849199	694,685,069,336.95	38,900.00

Note: The Optimum Value is -4,000,263,000,000.

[1] % Diff. is (Minimum Obj. Value – Optimum Value)/Optimum Value x 100%

[2] Value Diff. is |Minimum Obj. Value – Optimum Value|

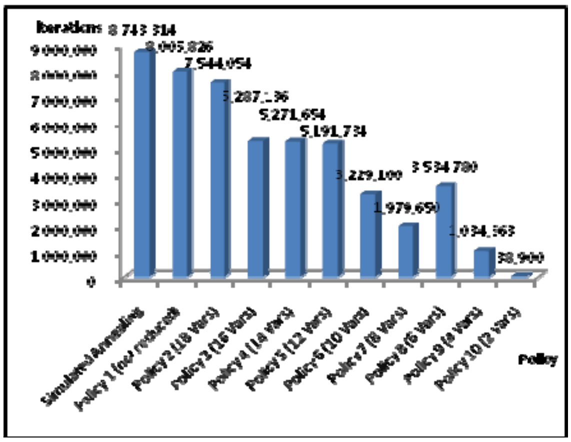

Fig 3. Iteration results: identification of policy

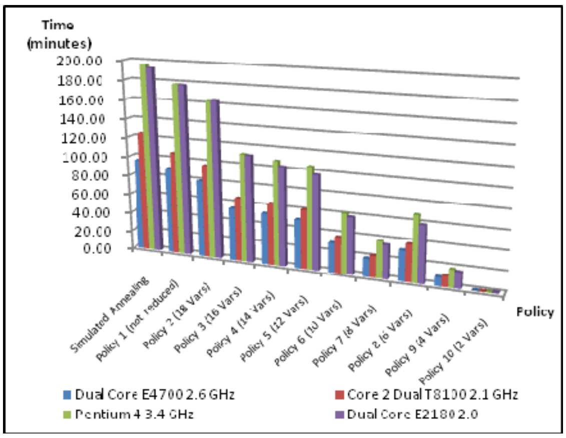

Fig 4. Time to run optimization by same seed: CPU benchmarks

REFERENCES

1. James C. Spall (2003). *Introduction to stochastic search and optimization: estimation, simulation, and control.* Wiley, New Jersey.
2. Susan M. Sanchez , Farhad Moeeni and Paul J. Sanchez (2006)., *So many factors, so little time… Simulation experiments in the frequency domain*, Science Direct , Int. J. Production Economics 103, 149-165.
3. Douglas J. Morrice (1989). *Simulation sensitivity analysis using frequency domain experiments.* Proceedings of the 1989 Winter Simulation Conference.
4. Schruben, L. W. And V. J. Cogliano (1987). *An Experimental Procedure for Simulation Response Surface Model Identification. Commu. Association for Computing Machinery, 30, 716-730.*
5. P. J. M. van Laarhoven and E. H. L. Aarts (1987). *Simulated Annealing Theory and Applications*. Springer.
6. Jacobson, S., A. Buss and L. Schruben (1991). *Driving Frequency Selection for Frequency Domain Simulation Experiments.* Operations Research, 39(6), 917-924.