Response Surface Optimization via Steepest Ascent, Simulated Annealing and Ant Colony Optimization Algorithms

LUANGPAIBOON P. and SERMPATTARACHAI P. [1)]

[1)] Department of Industrial Engineering, Faculty of Engineering,
Thammasat University, 12120 THAILAND
(662)-5643002-9 ext. 3081 Fax (662)-5643017, Email: lpongch@engr.tu.ac.th

Abstract

Nowadays, many entrepreneurs face to extreme conditions for instances; costs, quality, sales and services. Moreover, technology has always been intertwined with our demands. Then almost manufacturers or assembling lines adopt it and come out with more complicated process inevitably. At this stage, products and service improvement need to be shifted from competitors with sustainability. So, a simulated process optimization is an alternative way for solving huge and complex problems. Many researchers suggested applying the problems with heuristic methods, such as Steepest Ascent, Simulated Annealing and Ant Colony Optimization algorithms. The first one is a conventional evolutionary operation to improve a process yield. The second is a process of classical statistical mechanics. Finally, Ant Colony Optimization imitates the real ant activities for solving any problems which be done by a simple communication or pheromone among ants.

In this research, tested problems are formulated into mathematical models with and without noises which are representing for ideal problems and easily illustrated by Response Surface Methodology. Heuristic algorithms have then been developed to solve these problems through a computer simulation program. In addition, a proposed algorithm is implemented to an industrial problem in order to find the maximal force of a spring model via a parameter adjustment. From experiments, the combined algorithm of Simulated Annealing and Ant Colony Optimization seems to work more properly in both cases of simulated processes and the spring force system. However, Ant Colony Optimization algorithm can search for the better yield with the same number of experimental runs.

1. Introduction

Response Surface Methodology (RSM) is a bundle of mathematical and statistical techniques that are helpful for modeling and analyzing problems. A response of our interest is influenced by several predictor variables. An objective is to optimize this response. For example, suppose that a process engineer wishes to find the levels of temperature (x_1) and pressure (x_2) that maximize the yield (y) of a process. The process yield is a function of levels of temperature and pressure.

$$y = f(x_1, x_2) + \varepsilon \qquad (1)$$

Where ε represents the level of noise (standard deviation) or error observed in the response y. If we denote the expected response by $E(y) = f(x_1, x_2) = \eta$, then the surface represented by

$$\eta = f(x_1, x_2) \qquad (2)$$

So, it is called a response surface.

We usually represent the response surface graphically, such as in Fig. 1, where η is plotted versus the level of x_1 and x_2. To help visualize the shape of a response surface, we often plot the contours of the response surface as shown in Fig. 1. In the contour plot, lines of constant response are drawn in the x_1-x_2 plane. Each contour corresponds to a particular height of the response surface.

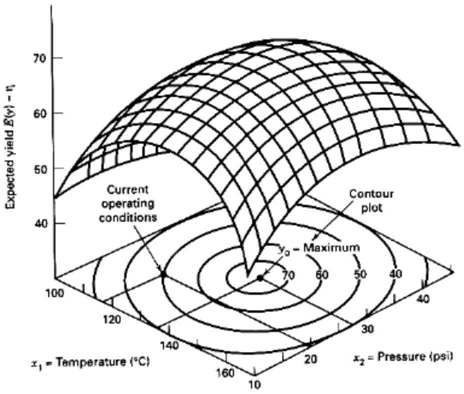

Fig. 1 A three dimensional response surface showing the expected yield with a contour plot of a response surface [9].

A response surface above describes how the yield of a process varies with changes in k independent variables. Estimation of such surfaces, and hence identification of near optimal settings for predictor variables is an important practical issue with interesting theoretical aspects. Many systematic methods for making an efficient empirical investigation of such surfaces have been proposed in the last fifty years. These are generally referred to as evolutionary operation (EVOP). RSM is used to improve the current operating conditions until the conditions of optimal yield are satisfied. In most RSM problems, a form of the relationship between the response and the independent variables is unknown. Thus, the first step in RSM is to find a suitable approximation for the true functional relationship between y and the set of its independent variables. Usually, a low-order polynomial in some region of the independent variables is employed [2]. If the response is well modeled by a linear function of the independent variables, then the approximating function is the first-order model.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon \qquad (3)$$

If there is curvature in the system, then a polynomial of higher degree must be used, such as the second-order model.

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum\sum_{i<j} \beta_{ij} x_i x_j + \varepsilon \qquad (4)$$

Almost RSM problems use one or both of these models. Of course, it is unlikely that a polynomial model will be a reasonable approximation of the true functional relationship over the entire space of the independent variables, but for a relatively small region they usually work quite well. The response surface analysis is then performed using the fitted surface. If the fitted surface is an adequate approximation of the true response function, then analysis of the fitted surface will be approximately equivalent to analysis of the actual system. The model parameters can be estimated most effectively if proper experimental designs are used to collect the data. Designs for fitting response surfaces are called response surface designs. RSM is a sequential procedure. Often, when we are at a point on the response surface that is remote from the optimum, such as the current operating conditions in Fig. 2, there is little curvature in the system and the first-order model will be appropriate.

An objective of this research is to lead the experimenter rapidly and efficiently along a path of improvement toward the general vicinity of the optimum. From Fig. 2, we see that the analysis of a response surface can be thought of "climbing a hill," where the top of the hill represents the point of maximum response. If the true optimum is a point of minimum response, then we may think of "descending into a valley". A simulation study is based on the function of process variables with different levels of noise. Response surface functions include different turning machining and spring force models. The objective of using the integrated approach is to find the values of the process variables which give the greatest yield, and to find these values with a minimum number of process runs at sub-optimal conditions. Conclusions are drawn, and practical recommendations are made.

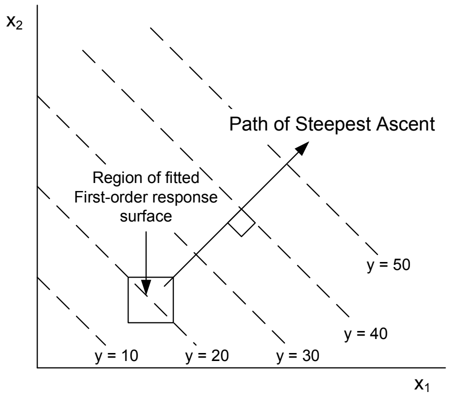

Fig. 2 The sequential nature of RSM [9].

2. Test Problems

In this paper, three algorithms consist of Steepest Ascent (Steepest), Simulated Annealing (SA) and Ant Colony Optimization (ACO). They will be used to operate and analyze the results under various types of mathematical functions (Table 1). The functions will be used as simulated processes with 2-5 predictor variables.

Table 1 Details of function types and their mathematical models

Function	Variable	Formula
Branin	2	$f(x)=5-\log_{10}[(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+(10-\frac{5}{4\pi}\cos(x_1))+10]$
Camelback	2	$f(x)=10-\log_{10}[x_1^2(4-2.1x_1^2+\frac{1}{3}x_1^4)+x_1x_2+4x_2^2(x_2^2+1)]$
Goldstein-Price	2	$f(x)=10+\log_{10}[1/\{1+(1+x_1+x_2)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\}*$ $\{30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\}]$
Parabolic	k	$f(x)=12-\sum_{j=1}^{k}[(-x_j)^2/100]$
Rastrigin	k	$f(x)=80-[20+\sum_{i=1}^{n}x_i^2-10(\sum_{i=1}^{n}\cos 2\pi x_i)]$
Rosenbrock	k	$f(x)=70[(\{20-((-x_1/a_1)^2+\sum_{j=2}^{k}[(x_j/a_j)-(x_1/a_1)^2]^2)\}+150)/170]+10$
Shekel	k	$f(x)=100\sum_{i=1}^{n}\frac{1}{c_i+\sum_{j=1}^{k}(x_j-a_{ij})^2}$
Styblinski	k	$f(x)=275-[(\frac{(x_1^4-16x_1^2+5x_1)}{2})+(\frac{(x_2^4-16x_2^2+5x_2)}{2})+\sum_{i=3}^{k}(x_i-1)^2]$

3. Spring Force Problem [11]

In this section, we propose a spring force model we have used in the performance study. The mathematical model is defined to maximize a spring force which reacts to spring conditions or parameters.

$$Max.\,Y=(300+16x_5)\left(\frac{140}{x_1}-1\right)+x_3\left(x_2+(x_5-20)\left(\frac{280}{x_1}-1\right)-x_4\right)\left(\frac{280}{x_1}-1\right) \quad (43)$$

; where X_1 = Edge of paper which faces to shaft (Values from 100-180)
X_2 = Joint of spring (Values from 35-75)
X_3 = Strength of spring (Values from 5-15)
X_4 = Compression distance of spring (Values from 20-50)
X_5 = Thickness of paper (Values from 0-50)

4. Methods

In this section, we describe how to find the best solution among different methods, i.e. Steepest Ascent, Simulated Annealing and Ant Colony Optimization algorithms.

4.1 Steepest Ascent Algorithm

The variation [5] is a 2^k factorial design with an additional reference point at the centre. The data from these design points are analysed. If there is evidence of a main effect, at some chosen level of statistical significance, and no evidence of curvature, at the same level of significance, another 2^k design with an additional reference point at the centre is carried out, centred on the point with the highest yield. It is possible, although rather unlikely, that this point is the centre of the preceding design. If there is no evidence of a main effect or of curvature the design is replicated. The details of the algorithm (Fig. 3) follow.

1. Define an objective for an optimization (Maximization or Minimization) of tested problems; for instance, Branin Surfaces, Camelback Surfaces, Goldstein-Price Surfaces, Parabolic Surfaces, Rastrigin Surfaces, Rosenbrock Curved Ridge Surfaces, Shekel Multi Peak Surfaces and Styblinski Surfaces.
2. Random a starting point as center point of a factorial design.
3. Calculate a response (y) for each point of the factorial design which compose of a center and peripheral point. Then formulate a first order model.
4. Calculate $\beta_0, \beta_1, \ldots, \beta_k$ by the least square method from the first order model or a linear regression.

$$Y = \beta_0 \dot{+} \beta_1 X_1 \dot{+} \beta_2 X_2 \quad (44)$$

Or calculate response Y from

$$Y = (\Sigma Y\ldots)/N + (EFFECT\ X_1 / 2)\ X_1 + (EFFECT\ X_2 / 2)\ X_2 \quad (45)$$

5. Review the suitable of the first order model by looking at each of linear regression coefficient (β_i). If none of linear regression coefficient is equal to zero, all factors are significant to the model.
6. Redo the same procedure; otherwise test a quadratic effect, in case of an unsuitable equation.
7. 7.1 If model is suitable, move a center coordinate ($x_1, x_2, \ldots, x_k$) to a new coordinate ($x_1^N, x_2^N, \ldots. x_k^N$) by calculating a step size (ΔX_i) which is related to the following equation:

$$\Delta X_i = \beta_i / (\beta_{Largest} / \Delta X_{Largest}) \quad (46)$$

Then calculate a new coordinate from $X_i^N = X_i + \Delta X_i$
7.2 Scale with a multiplication of ‘n’ where n = 1, 2, … until a response (Y_n) could not get a better value then termination.

$$Y_n = Origin + n\Delta \quad (47)$$

8. Repeat 4-7 to calculate the responses.
9. Compare responses of each iteration and keep the best value for a solution.
10. Terminate when the criteria is met.

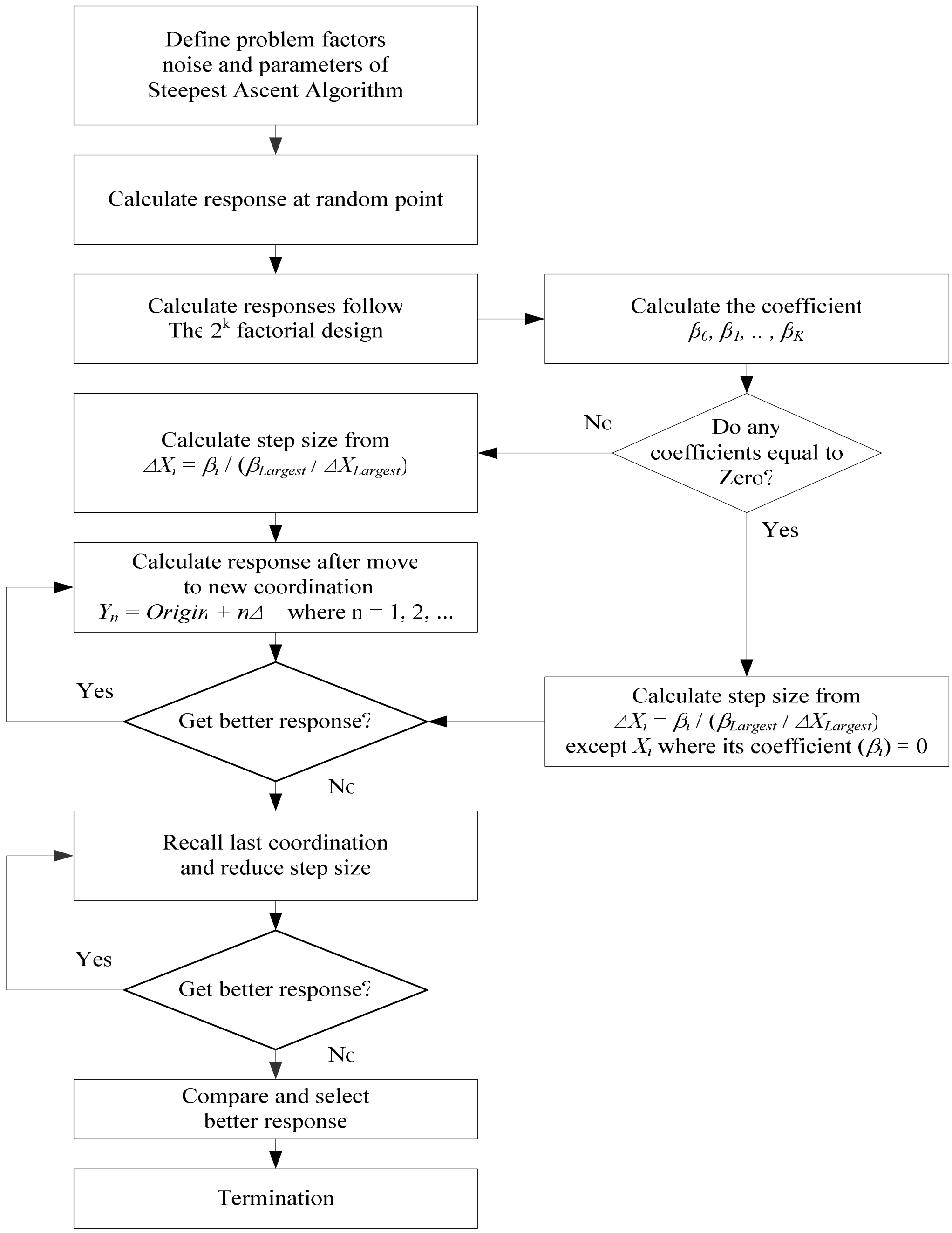

Fig. 3 Flow Chart of Steepest Ascent Algorithm

4.2 Simulated Annealing Algorithm

Kirkpatrick et al. [7] first proposed a detailed analogy of an annealing in solids to the combinatorial optimization. They imitated a framework for the optimization problems which are vary large and complex. Statistical mechanics generate the general discipline of condensed matters [3]. A fundamental question in statistical mechanics then concerns that the atoms remain the fluid or solid matter when the level of temperature approaches the "ground state" or the lowest energy state of the system. Ground states are extremely rare at the elevated temperatures, but are predominated properties at low temperatures. However, the low temperature is not sufficient to determine the ground states [12]. In practice, care must be taken at the stage of an annealing. This would allow the system reaches the ground states. The annealing processes are performed by first melting the system at a high temperature, then lowering the temperature slowly, finally spending a long time at freezing temperatures. During the annealing process, the time spent at each temperature level must be sufficiently long to allow the system to reach a thermal equilibrium or a steady state. If care is not taken in adhering to the annealing temperature schedule, undesirable random fluctuations may cause the shift of the ground state. The basic idea of statistical mechanics initiates a generalization of the iterative improvements or the search for a better solution of the combinatorial optimization. This process is encountered to a 'Steepest-Descent' algorithm for Minimization problems or a 'Steepest-Ascent' algorithm for maximization problems.

In this paper, Simulated Annealing algorithm (Fig. 4) is programmed with Microsoft Visual Basic for Application. When select tested problem and noise, the algorithm calculates the responses (Y_i) and variables (X_i) with following steps.

1. Define an objective for an optimization (maximization or minimization) of the tested problems as appeared in the details of Steepest Ascent algorithm.
2. Define parameters of the algorithm; i.e. a starting temperature, a freezing temperature, a reducing rate and iteration.
3. Random a starting point with 'k' variables related to size of the tested problems. Then calculate the responses (Y_i) by replacing variables (X_1, X_2, … X_k) with random values as a starting point (s).
4. Create an anneal schedule from a starting temperature and a reducing rate; i.e. the starting temperature is 2°C, the reducing rate is 0.9 then the next anneal temperature is equal to 1.8°C.
5. Set a counter for iteration in each anneal temperature.
6. Calculate the neighbor responses (s_n) from the anneal schedule by replacing a variable (X_i) of the tested problem.
7. Compare the calculated responses in step 3 and 6 then calculate a value of ΔE by s_n - s. In case of Maximization, if ΔE is grater than '0', replacing variables at condition s with s_n.
8. On the other hand, if the calculated value of ΔE is less than '0', random the probability number (q_1) to compare with Boltzman value (q_0) where Boltzman (q_0) = $exp(-\Delta E/K_b t)$
9. If the probability number (q_0) is greater than Boltzman value (q_1), replacing variables at condition s with s_n.
10. Do step 6-9 until an iteration criterion is reached.
11. Calculate next anneal temperature as describe in step 4.
12. Do step 5-11 until the anneal temperature is less than the freezing temperature. Then a termination criterion has been reached.

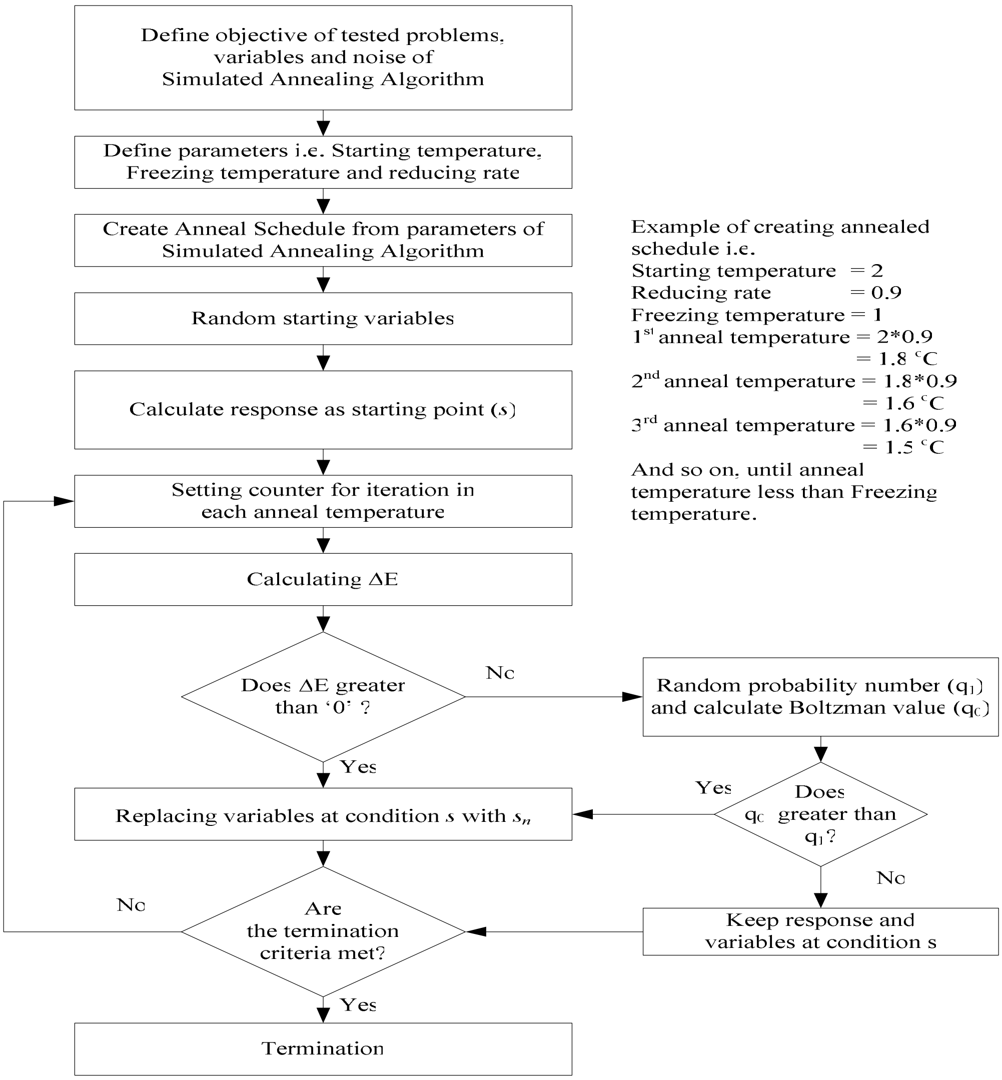

Fig. 4 Flow Chart of Simulated Annealing Algorithm

4.3 Ant Colony Optimization Algorithm

Ant algorithm was first proposed by Dorigo and his colleagues as a multi-agent approach to optimization problems, such as a traveling salesman problem (TSP) and a quadratic assignment problem (QAP). There is currently a lot of ongoing activity in the scientific community to extend or apply ant-based algorithms to many different discrete optimization problems. Recent applications cover problems like a vehicle routing, a plant layout and so on.

Ant algorithm is inspired by observations of real ant colonies. Ants are social insects and they live in colonies. A behavior is direct more to the survival of the colony as a whole than to that of a single individual component of the colony [8, 10]. Social insects have captured the attention from many scientists because of a structure of their colonies, especially when compared with a relative simplicity of the colony's individual. An important and interesting behavior of ant colonies is their foraging behavior and in particular how ants can find shortest paths between food sources and their nest [1].

While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. Ants can smell pheromone and when choosing their way. They tend to choose, in probability paths marked by strong pheromone concentrations. The pheromone trail allows the ant to find their way back to the food source (or to the nest). Also, it can be used by other ants to find the location of the food sources found by their nest mates [4]. Ant Colony Optimization algorithm (Fig. 5) consists of the iteration steps where each ant makes its own solution as follows.

1. Define an objective for an optimization (Maximization or Minimization) of the tested problems as of the tested problems as appeared in the details of Steepest Ascent algorithm.
2. Define parameters for Ant Colony Optimization Algorithm, such as number of ants and moves etc.
3. Each ant make its own initial states (s), paths and communicate the responses and coordinates where
   - Construct the feasible solution.
   - Evaluate the generated solution.
   - Decide to retrace the path that the ant has followed.
4. Random 'k' variables ($X_1, X_2, \dots X_k$) for initial states (s) of each ant which turn on the ant activities and compare its responses and termination criteria.
5. From Initial state (s), ant activities drive all ants in system and move to its neighborhood state (s_r).
6. While each ant locates at neighborhood states (s_r), a system compares the responses and its initial states (s). If any response of the same ant is better than its initial states (s), then move to a neighborhood state (s_n).
7. In case of neighborhood states (s_n) less than the previous state, the system generates a probability number (q_1) and compare with a certain number (q_0). If q_1 is greater than q_0, a movement of each ant is going ahead. Otherwise, no movement.
8. In case of no better neighborhood response, set this state as 'Local Optima' (L_i) and wait for a communication from other ants at other 'Local Optima' (L_j).
9. Compare among 'Local Optima' ($L_1, L_2, \dots, L_i, L_j, \dots, L_n$) and set a direction of the path to the best Local Optima.
10. Construct the solution by repeating steps 5-10, until the termination conditions are met.

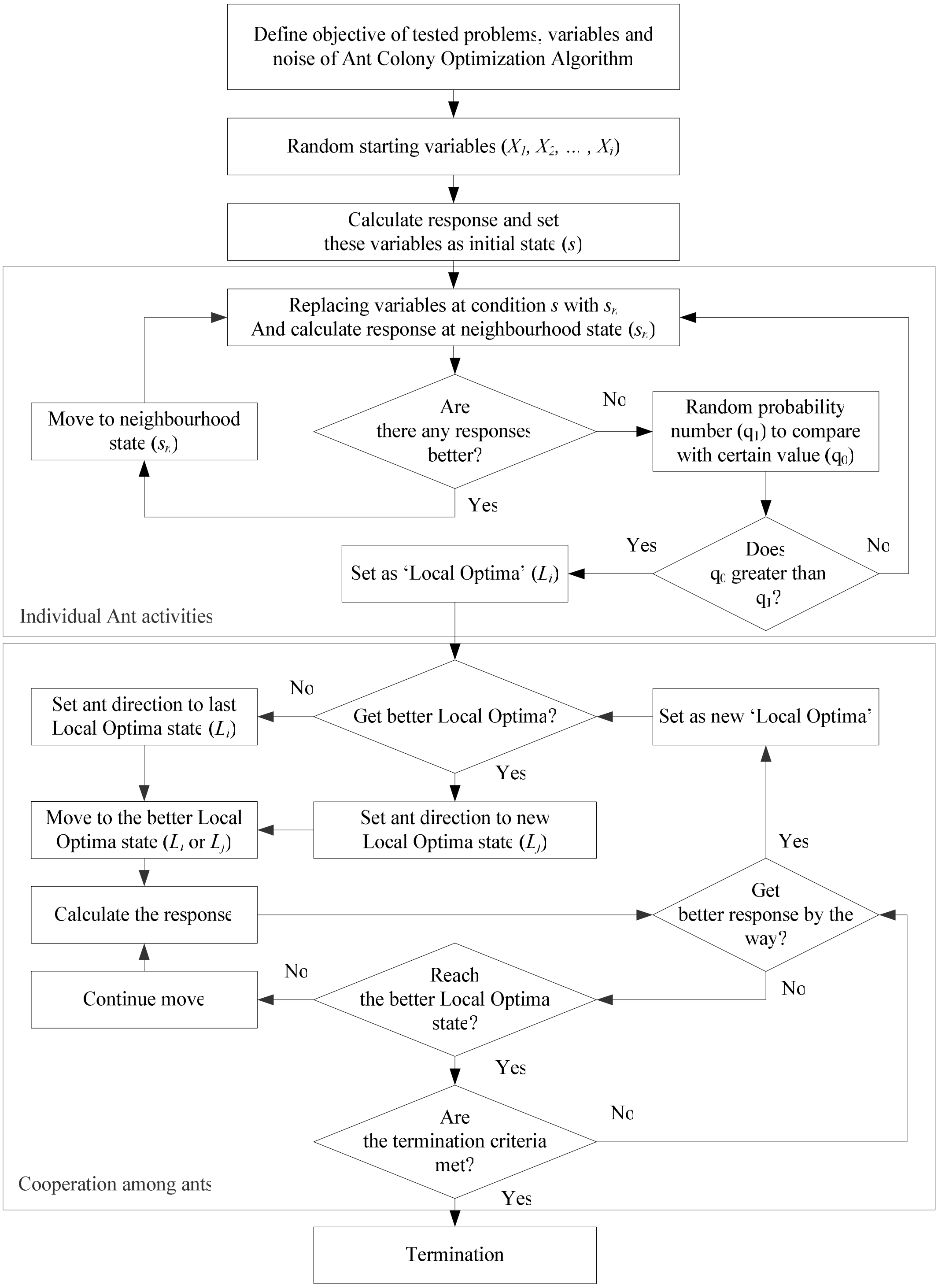

Fig. 5 Flow Chart of Ant Colony Optimization Algorithm

5. Experimentation

In this section, we run the experiments and arrange all data into 2 categories. Firstly, the experiments are tested with tested problems which have 2-5 variables. The second experiments are applied to industrial problems. The optimal solution is defined by reviewing results from these performance measures, i.e. a mean or average response, standard deviation (SD) and S/N ratio. The S/N ratios depend on an objective of a problem (Table 2). All models were tested with the proposed algorithms of 15 replications to check a consistency of results on these three measures. The results are presented in the form of tables and figures with optimal conditions and response values. All the results are discussed in the next section. The all programs were developed on a PC using VBA (MS-Excel).

Table 2 Optimization methods and S/N ratio formula

Methods	S/N ratio formula	Objective
Larger is better	$S/N = -10*\log(\Sigma(1/Y^2)/n)$	Maximization
Smaller is better	$S/N = -10*\log(\Sigma(Y^2)/n))$	Minimization

5.1 Experiments with tested problems

The comparisons are made for four different levels of measurement noise on the response. There are 15 realizations in each experimental level of measurement noise. The noise is taken to be independently and normally distributed with mean of zero and standard deviations of 0, 1, 2 and 3. The experimental results are shown in Table 3.

Table 3 Summary of Mean, Standard Deviation and S/N ratio of tested problems

Model	Algorithm	2 Variables			3 Variables			4 Variables			5 Variables		
		Average responses	SD	S/N ratio	Average responses	SD	S/N ratio	Average responses	SD	S/N ratio	Average responses	SD	S/N ratio
Branin	Steepest	9.285	2.813	18.936	-	-	-	-	-	-	-	-	-
	SA	8.733	2.547	18.432	-	-	-	-	-	-	-	-	-
	ACO	10.412	3.360	19.845	-	-	-	-	-	-	-	-	-
Camelback	Steepest	15.107	1.560	23.487	-	-	-	-	-	-	-	-	-
	SA	13.110	1.922	22.201	-	-	-	-	-	-	-	-	-
	ACO	42.423	7.763	32.181	-	-	-	-	-	-	-	-	-
GoldsteinPrice	Steepest	9.627	1.579	19.496	-	-	-	-	-	-	-	-	-
	SA	8.130	1.638	17.839	-	-	-	-	-	-	-	-	-
	ACO	13.319	3.316	22.198	-	-	-	-	-	-	-	-	-
Parabolic	Steepest	17.016	3.933	24.378	16.575	3.675	24.171	16.108	3.368	23.948	15.668	3.115	23.725
	SA	16.667	3.660	24.221	16.213	3.515	23.989	15.779	3.125	23.785	15.333	3.026	23.535
	ACO	17.027	3.913	24.382	16.648	3.498	24.227	16.578	3.396	24.198	16.575	3.513	24.182
Rastrigin	Steepest	100.434	1.708	40.035	112.994	3.312	41.052	117.996	7.979	41.385	113.096	13.380	40.904
	SA	96.939	2.438	39.724	98.831	7.677	39.830	86.722	12.226	38.535	69.230	21.196	35.509
	ACO	104.165	3.399	40.349	122.526	3.676	41.759	139.348	4.427	42.875	156.568	5.202	43.884
Rosenbrock	Steepest	85.149	3.950	38.594	84.277	3.341	38.507	83.562	2.921	38.434	83.015	2.644	38.378
	SA	84.494	3.477	38.529	83.563	2.908	38.435	82.870	2.758	38.363	81.971	2.297	38.269
	ACO	84.808	3.662	38.560	84.727	3.624	38.552	84.630	3.590	38.543	84.490	3.471	38.528
Shekel	Steepest	21.541	2.270	26.608	18.347	1.574	25.201	14.700	2.731	23.054	12.238	3.977	21.127
	SA	20.783	1.848	26.309	16.028	1.905	23.954	11.656	2.528	21.002	8.884	2.717	18.466
	ACO	23.508	3.459	27.328	22.605	3.422	26.946	20.628	3.025	26.072	17.820	3.409	24.574
Styblinski	Steepest	356.327	2.619	51.037	392.825	1.613	51.884	386.950	4.303	51.752	375.259	6.234	51.484
	SA	352.354	2.469	50.939	374.117	8.395	51.454	355.716	12.392	51.009	329.166	20.275	50.304
	ACO	357.920	3.439	51.075	395.961	4.775	51.952	395.175	4.734	51.934	392.570	7.528	51.875

From results above we can conclude the characteristics, an advantage and a disadvantage in each method (Table 4). Most of responses from Ant Colony Optimization algorithm are quite close to the optimal and tolerance to various conditions when compared. ACO is then selected to determine the performances of an industrial problem.

Table 4 Advantage and disadvantage of solved methods against tested problems

Methods	Advantage	Disadvantage
Steepest Ascent	• Computation time at low level of noise is quite fast. • Data distribution is good at no noise condition. • Responses are quite good at no noise condition. • It's suitable for 2-3 variable problems.	• Computation time related to level of noise and complexity of problems. • Data distribution is quite spread out, low tolerance to noises and complexity problems. • Responses are fair at different noises and variables (over 4 variables).
Simulated Annealing	• Computation time is fastest and almost constant although conditions are changed at noises. • Data distribution is quite low and tolerate to noise. • Complexity of algorithm is low because of pure random method with certain terminate condition.	• Responses are fair at different noise and variables condition.
Ant Colony Optimization	• Computation time at low level of noise is quite low. • Data distribution is lowest at any conditions. • Response is closest to Global point, tolerate to noises and variables.	• Computation time related to noises and variables. When increase noises and variables in system, computation time is taken longer time to solve because of complexity of algorithm (ant activities and communications) to check 'Local optima'.

5.2 Experiments with an industrial problem through Ant Colony Optimization algorithm

This section presents the performance study of the industrial problem (Table 5). A spring test is studied in different conditions of independent factors such as a joint, strength and a compression distance to maximize the spring force. Ant Colony Optimization algorithm is applied to find the best response for maximizing force from a spring design.

Table 5 Results of a spring force problem through various Algorithms

Performance measures	Steepest Ascent			Simulated Annealing			Ant Colony Optimization		
	Run	**Time**	**Response**	**Run**	**Time**	**Response**	**Run**	**Time**	**Response**
Average	6766	0:00:13	2567.815	6000	0:00:22	2580.904	1968640	0:11:47	3166.560
SD	36	0:00:00	145.743	0	0:00:00	180.135	927	0:00:10	125.030
Max	6827	0:00:14	2806.2	6000	0:00:22	2948.350	1970073	0:12:07	3368.710
Min	6697	0:00:13	2335.007	6000	0:00:21	2331.012	1967196	0:11:30	2869.090
S/N ratio	-	-	68.154	-	-	68.183	-	-	69.992

5.3 Experiments on an industrial problem through a combined algorithm.

A combined algorithm of Simulated Annealing and Ant Colony Optimization is proposed to eliminate a disadvantage of the computation time. We present the performance study and evaluation of the industrial problem as illustrated in previous section. The results are listed in table below.

Table 6 Results of a spring force problem through a Combined Algorithm

Performance measures	Spring force problem		
	Run	**Time**	**Response**
Average	7393	0:00:23	2604.846
Std. Dev.	470	0:00:00	162.504
Max	8165	0:00:23	2926.799
Min	6750	0:00:22	2372.605
S/N ratio	-	-	68.281

5.4 Analysis of Data

From the results in sections 5.2-5.3, we found that Ant Colony Optimization algorithm contributed the best solution for the industrial problem. All operating costs from Ant Colony Optimization are lower than the combined algorithm that describe in the figure of box-plots below.

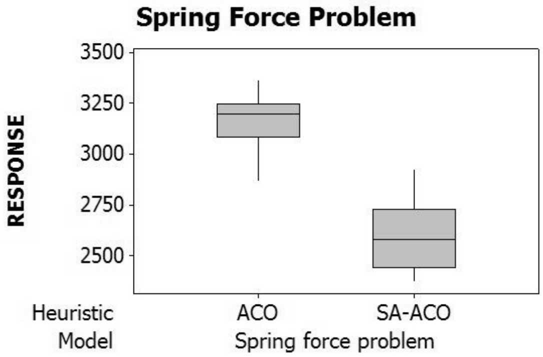

Fig. 6 Box plot analysis of the maximal force of a spring against the different algorithms

Table 7 Advantage and disadvantage of proposed method against a spring force problem

Proposed Methods	Advantage	Disadvantage
Ant Colony Optimization Algorithm	• Computation time for searching the best response in constrained problems is quite excellent (fastest). • Data distribution of response is quite excellent. • Response from all problems is better than the rest.	• Computation time for searching the best response of unconstrained problems take a long time.
Combined Algorithm	• Computation time for searching the best response of unconstrained problems is quite excellent because amount of sampling is reduced from combined algorithm.	• Response is poorer than Ant Colony Optimization algorithm.

In conclusion, we notice that the combined method which is developed from Simulated Annealing and Ant Colony Optimization algorithms enables to search the optimal response of unconstrained problems faster. That is the strong point of Simulated Annealing algorithm. However, it has to trade off searching ability for the optimal response with the computation time. The selection of the suitable method based on the types of problems should be considered carefully.

6. Conclusion and Discussion

From the overall results of experiments with tested problems, we found significant characteristics for performace measures which consist of an average value of responses, a standard deviation and S/N ratio. We can explain in each category below.

Problem Type (Function) – For a few variables (2-variable functions) and no noise, function types mainly are not affected to the methods. The computation time and responses are merely the same. In case of an industrial problem, the combined method works faster but the best solution can be achieved by Ant Colony Optimization algorithm.

Noise Level – Steepest ascent algorithm seems to be more efficient, in terms of speed of convergence, when the standard deviation of the noises is low. When we add noises (noise level from 1 to 3) into function, we noticed that the computation time are dramatically increased but the levels of noise do not significantly make different in the computation time. However, the response value is higher when noise levels are increased.

Amount of Variables or Size of Tested Problems – the best solutions from tested problems with 2-5 variables are obtained by Ant Colony Optimization algorithm. However, it can be clearly stated when variables are over 4 factors. Moreover, the effect of variables to the computation time is also increased significantly except Simulated Annealing algorithm.

The suggestion from the experiments, we observed that the weakest point of Ant Colony Optimization algorithm is the computation time for searching the best response. Hence, we tried to combine with Simulated Annealing algorithm to eliminate that weakness. However, results in the last experiment indicate that there is only a success in reducing the computation time for unconstrained problems. We have to trade off it with the performance measures for the best algorithm. From the results of the steepest ascent algorithm, percentage of sequences ended at the optimum or near optimum is slightly good at higher levels of error standard deviation although the greater number of runs were required to converge to the optimum. As stated earlier, the function of this research was restricted to proposed process variables. Consequently, comparisons and conclusions between the algorithms may not be valid for other families of functions. Other stochastic approaches could be extended to the steepest ascent algorithm based on conventional factorial designs to increase its performance. Finally, the combined algorithm takes longer computation time than its original algorithm on constrained problems. The applications on other processes could be determined to confirm the performance [6].

7. References

[1] Bell, J. E., and McMullen, P. R.; Ant Colony Optimization Techniques for the Vehicle Routing Problem, Advanced Engineering Informatics, Vol. 18, pp. 41-48, 2004.

[2] Box, G. E. P., and Draper, N. R.; Evolutionary Operation: A Statistical Method for Process Improvement, John Wiley and Sons, New York, pp. 19-69, 1969.

[3] Golden, B. L., and Skiscim, C. C.; Using Simulated Annealing to Solve Routing and Location Problem, Working Paper Series of University of Maryland, 1984.

[4] Goss et al., Self-Organized Shortcuts in the Argentine Ant, Naturwissenschaften, Vol. 76, pp. 579-581, 1989.

[5] Jalayanadecha, A. and Luangpaiboon, P.; Process Improvement via Evolutionary Operation Techniques: Simplex and Steepest Ascent, Proceedings of the National Industrial Engineering Network Conference, Phitsanulok, Thailand, pp. 177 – 184, 2003.

[6] Khan, Z., Prasad, B., and Singh, T.; Machining Condition Optimization by Genetic Algorithms and Simulated Annealing, Computers Ops Res, Vol. 24, pp. 647-657, 1997.

[7] Kirkpatrick, S., Gelatt, C. D. Jr., and Vecchi, M. P.; Optimization by Simulated Annealing, Science, 1983.

[8] McKendall, A. R. Jr., Shang, J., and Kuppusamy, S.; Hybrid Ant Systems for the Dynamic Facility Layout Problem, Computer & Operations Research, pp. 1-14, 2005.

[9] Montgomery, D. C.; Design and Analysis of Experiments, 5th edition, USA, John Wiley and Sons, 2001.

[10] Solimanpur, M., Vrat, P., and Shankar, R.; An Ant Algorithm for the Single Row Layout Problem in Flexible Manufacturing Systems, Computers & Operations Research, Vol. 32, pp. 583-598, 2005.

[11] Sudsana-na-Ayudthya. P., and Matakangana A.; Comparison of Non-Linear Programming without Derivatives and Response Surface Methodology for Optimal Parameter, Proceedings of Seminar in Applied Optimization, 2001.

[12] Tang O.; Simulated Annealing in Lot Sizing Problems, International Journal of Production Economics, Vol. 88, pp. 173–181, 2004.