

# Process Optimisation via Firefly and Ant Colony Optimisation Elements on the Path of Steepest Ascent for a CSTR

Pongchanun Luangpaiboon, *Member, IACSIT*

**Abstract**—The objective of this work is to make use of conventional response surface methodologies and basic elements from metaheuristic algorithms in the design of influential variables for engineering systems. A method of steepest ascent and integrated approaches of the method of steepest ascent with firefly and ant colony optimisation algorithms, are compared on a simulated continuous stirred tank reactor or CSTR with various levels of signal noise. The experimental results suggest that the conventional method of steepest ascent seems to be the most efficient on the CSTR surface at the lower levels of noise. However, the integrated approaches with both firefly and ant colony optimisation elements work well when the standard deviation of the noise is at higher levels. Although the average, the standard deviation of the greatest actual concentration of the product and percentage of sequences ended at the optimum from the integrated algorithm are better, they need more average design points, especially with ant colony optimisation element, to converge to the optimum when compared.

**Index Terms**—Firefly, ant colony optimisation, steepest ascent, continuous stirred tank reactor.

## I. INTRODUCTION

The objective of response surface methodology is to describe how the process response varies with changes in  $k$  predictor variables. Estimation of such response surfaces, and hence identification of near optimal settings for process variables is an important practical issue with interesting theoretical aspects. The predictor variables determined will count on the specific field of the application. Most industrial processes have some predictor variables. These predictor variables can be changed by plant operators or by automatic control mechanisms to improve the efficiency of the machine. One among those processes of chemical reactor is the most influential and therefore important unit to be encountered by a chemical engineer. Non-linear and linear descriptions are derived to suit the system. A widely used type of reactor in various industries is a well-stirred tank into which there is a continuous flow of reacting material and from which the reacted or partially reacted material passes continuously. To

ensure the successful operation of a continuous stirred tank reactor (CSTR) it is necessary to understand its dynamic characteristics.

The aim of this paper is to introduce some basic concepts of chemical reaction system and develop computer simulation models for the CSTR. A good understanding will ultimately enable effective control systems design of its predictor variables via the conventional sequential algorithm of the steepest ascent (SAM). There is much current interest in optimisation methods with the stochastic elements. Natural intelligence-inspired approximation optimisation techniques called metaheuristics are then introduced to the SAM. Generally, metaheuristics have been used to avoid being trapped in local optima with a poor value [1]. The common factor in metaheuristics is that they combine rules and randomness to imitate natural phenomena. They widely grow and apply to solve many types of problems.

The major reason is that metaheuristic approaches can guide the stochastic search process to iteratively seek near optimal solutions in practical and desirable computational time. Their properties expose useful information and overcome the large and noisy systems such as the CSTR. These metaheuristics are then received more attention in the last few decades. They can be categorised into three classes: biologically-based inspiration, e.g. genetic algorithm or GA [2], memetics algorithm or MAs [2], shuffled frog leaping algorithm or SFLA [2], firefly algorithm or FFA [3], bees algorithm or BEES [4], harmony search algorithm or HSA [5], neural network or NN [6], ant colony optimisation or ACO [7], evolutionary programming or EP [8], differential evolution or DE [9] and particle swarm optimisation or PSO [10]. Moreover, there are some with the socially-based inspiration, e.g. tabu search or TS [11] and the physically-based inspiration such as simulated annealing or SA [12].

The objective of this study is to compare the efficiency of sequential algorithms for on-line optimisation of a chemical process in the presence of noises. The method of steepest ascent and the integrated approach between the method of steepest ascent and two classes of metaheuristics, firefly and ant colony optimisation algorithms, are selected and implemented on the CSTR. The context is maximising the concentration of a desired product of a chemical reactor with respect to feed rate, concentration and temperature. The paper is organised as follows. Sections II and III are concerned with the related methods and main features of the dynamic model of chemical reactors, respectively. Section IV briefly discusses the proposed methods and some preliminary assumptions of process variable levels. Section V provides experimental results and discussions.

Manuscript received October 12, 2012; revised December 7, 2012. This work was supported by the Higher Education Research Promotion and National Research University Project of Thailand, Office of the Higher Education Commission. The authors wish to thank Thammasat University, THAILAND for the financial support.

Pongchanun Luangpaiboon is with the Industrial Statistics and Operational Research Unit (ISO-RU), Department of Industrial Engineering, Faculty of Engineering, Thammasat University, 12120, Thailand (tel.: (662)564-3002-9; fax: (662)564-3017; e-mail: lpongch@engr.tu.ac.th).

## II. RELATED METHODS

### A. Steepest Ascent Method (SAM)

The procedure of SAM is that a hyperplane is generated from the initial  $2^k$  factorial design and regression analysis. If there is an evidence of main effect(s), at some chosen level of statistical significance and no evidence of curvature, at the same level of significance, the direction of steepest ascent on the hyperplane is then determined by the principle of least squares. The next experimental run is carried out at a point, which has some fixed distance in this direction, and further runs are carried out by continuing in this direction until no further increase in yield is noted [1]. When the response first decreases and there is no improvement of two more verified yields, another  $2^k$  factorial design will be carried out, centered on the preceding design point. A new direction of steepest ascent is estimated from this latest experiment. Provided at least one of the coefficients of the hyperplane is statistically significantly different from zero, the search continues in this new direction (Fig. 1). Once the first order model is determined to be inadequate, the area of optimum is identified via a second order model or a finishing strategy.

#### Procedure of the SAM ()

```

While (termination criterion not satisfied) – (line 1)
  Initialise algorithm parameters;
  Define the objective function of  $f(\mathbf{x})$ , where  $\mathbf{x}=(x_1, x_2, \dots, x_k)^T$ ;
  Generate the initial factorial design point or  $\mathbf{x}_i$  ( $i=1, 2, \dots, 2^k$ );
  Determine the response function at  $\mathbf{x}_i$  via  $f(\mathbf{x}_i)$ ;
  Schedule activities (when Regression verification criteria not satisfy)
    Determine significant first order model from the factorial design points;
    Schedule activities
      Move along the steepest ascent's path with a step length ( $\Delta$ );
      Compute response functions;
      If new response function is greater than the preceding then
        Move ahead with another  $\Delta$ ;
      Else calculate two more response function to verify the descending trend;
        If one of which response function turn out to be greater than the preceding coordinate's response function;
          Then use the biggest response function to continually move along the same path;
        Else
          Use closest preceding point as a centre for the new  $2^k$  factorial design;
        End if;
      End if;
    End schedule activities;
  End while;
End procedure;

```

Fig. 1. Pseudo code of the SAM.

### B. Firefly Algorithm (FFA)

When you submit your final version, after your paper has been accepted, prepare it in two-column format, including figures and tables. The firefly algorithm is a metaheuristic algorithm, inspired by the flashing behaviour of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. Now this can idealise some of the flashing characteristics of fireflies so as to consequently develop various versions of the firefly-inspired algorithm. For simplicity in describing the firefly algorithm, there are three idealised rules as follows. On the first rule, each firefly

attracts all other fireflies with lower levels of flashes. All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex. Secondly, attractiveness is proportional to their brightness which is reverse proportional to their distances. For any two flashing fireflies, the less bright one will move towards the brighter one. Finally, if there is no brighter one than a particular firefly it randomly moves. For a maximisation problem, the brightness can simply be proportional to the value of the objective function. Based on these three rules, the basic steps of the FFA can be summarised as the pseudo code shown in Fig. 2.

#### Procedure of the FFA Metaheuristic()

```

Begin;
  Initialise algorithm parameters;
  Define the objective function of  $f(\mathbf{x})$ , where  $\mathbf{x}=(x_1, x_2, \dots, x_k)^T$ ;
  Generate the initial population of fireflies or design point or  $\mathbf{x}_i$  ( $i=1, 2, \dots, n$ );
  Determine the light intensity or response function of  $I_i$  at  $\mathbf{x}_i$  via  $f(\mathbf{x}_i)$ ;
  While ( $t < \text{MaxGen}$ )
    For  $i = 1$  to  $n$  (all  $n$  fireflies);
      For  $j = 1$  to  $n$  ( $n$  fireflies)
        If ( $I_j > I_i$ );
          Move firefly  $i$  towards  $j$ ;
        End if;
        Attractiveness varies with distance  $r$  via  $\text{Exp}[-\gamma r^2]$ ;
        Evaluate new solutions and update light intensity;
      End for  $j$ ;
    End for  $i$ ;
    Rank the fireflies and find the current best;
  End while;
  Postprocess results and visualisation;
End procedure;

```

Fig. 2. Pseudo code of the FFA metaheuristic.

### C. Ant Colony Optimisation (ACO)

Ant algorithm was first proposed by Dorigo and his colleagues as a multi-agent approach for solving optimisation problems, such as quadratic assignment (QAP) and travelling salesman problems (TSP). There is currently a lot of ongoing activity in the scientific community to extend or apply ant-based algorithms to various optimisation 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. Behaviour is direct more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention from many scientists because of their colony structure, especially when compared with a relative simplicity of the colony's individual. In ant colonies the foraging behaviour and in particular how ants can find shortest paths between food sources and their nest are very interesting. While walking from food sources to the nest and vice versa, ants deposit a substance called pheromone on the ground, forming a pheromone trail. With ant ability to smell pheromone they tend to choose a path marked by strong pheromone concentrations with the higher probability. The pheromone trail allows the ants to find their way back to the food source and vice versa. It can be also used by other ants to find the location of the food sources found by their nest mates. The pseudo code is used to briefly explain to all the procedures of ACO shown in Fig. 3.

**Procedure of the ACO Metaheuristic()**
**While** (termination criterion not satisfied) – (line 1)

Initialise algorithm parameters;

 Define the objective function of  $f(x)$ , where  $x=(x_1, x_2, \dots, x_k)^T$ ;

 Generate the initial design point or  $x_i$  ( $i=1, 2, \dots, n$ );

 Determine the response function at  $x_i$  via  $f(x_i)$ ;

**Schedule activities**

Ant generation and starting design points;

Makes path or step for each ant;

Compare response function;

**If** no improvement of response function **then**

Communication with best ant response function;

Make path or step from local trap to best ant;

**Else**
**If** ant found the better response function **then**

Go to line 5;

**Else**

Wait for best ant communication;

**End if**;

**End if**;

**End schedule activities**;

**End while**;

**End procedure**;

Fig. 3. Pseudo code of the ACO metaheuristic.

## III. CONTINUOUS STIRRED TANK REACTOR (CSTR)

For the CSTR a stream rich in chemical A of feed concentration  $C_{A(in)}$  is flowing into a reactor at a feed flow rate of  $F_{(in)}$ , and a feed temperature of  $T_{(in)}$ . The reaction in the CSTR is an irreversible, first order exothermic reaction. The proportion of chemical A is converted to a desired product B, which, in turn, at high temperature undergoes further reaction and is decomposed to form an undesired by-product C. The stated objective is to explore the operating conditions corresponding to higher concentration of product. It is also assumed that level is perfectly controlled, so the volume of material in the tank is constant. This implies that the flow out equals the flow in. The temperature in the reactor may be regulated by manipulating the flow rate of the cooling water ( $F_C$ ) in the heat exchanger (Fig. 4). A mechanistic model adequately accounting for the system under study is suggest purely by physical consideration and the dynamics of the system can then be described by the following set of ordinary, non-linear differential equations. There are three predictor variables, which can be set to any chosen values within safe limits. These predictor variables relate to the feed flow are shown in Table I. The response variable of the process is defined to be the concentration of the desired product B,  $C_B$ .

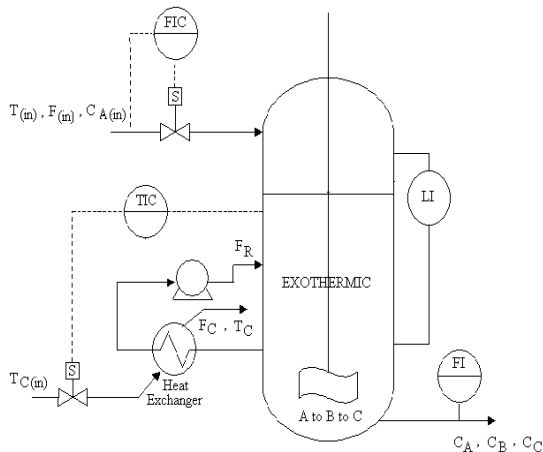


Fig. 4. A diagrammatic representation of a single continuous stirred tank reactor (CSTR).

TABLE I: PREDICTOR VARIABLES OF FEED FLOW AND THEIR SAFE LIMITS

Predictor variables	Description	Unit	Feasible region
$T_{(in)}$	Feed temperature of reactant A	Celsius	60-100
$F_{(in)}$	Feed flow rate of reactant A	Liter/min	1-10
$C_{A(in)}$	Concentration of reactant A	Mole/liter	1-15

## IV. DETAILS OF THE PROPOSED METHODS

**SAM Parameters:** the volume of the  $2^3$  factorial design [8.0]; the step length [1.0]; the significance level for tests of significance of slopes [10%].

**FFA Parameters:** the attractiveness at  $r=0$  of  $\beta_0$  [1.0]; the light absorption coefficient of  $\gamma$  [0.01].

**ACO Parameters:** pheromone evaporation rate of  $\rho$  [0.4]; heuristic factor for preferring among available options of  $\eta$  [0.5];  $\zeta$  and  $\delta$  are exponent parameters that control the relative importance of pheromone concentration versus the heuristic factor [0.5 and 2.5].

**Step 1:** Perform a  $2^3$  factorial design at a random centre design point.

**Step 2:** Fit a regression plane to all design points.

**Step 3:** Test whether there is evidence that either  $\beta_1$ ,  $\beta_2$  or  $\beta_3$  is different from zero at the 10% level of significance.

**Step 4a:** If the result is significant, move one step along the path of steepest ascent and determine the yield and go to **Step 5a**. Otherwise go to **Step 4b**.

**Step 4b:** Test whether there is evidence that the interaction or curvature check is significant. If the check is significant, go to **Step 6**. Otherwise, replicate the design and return to **Step 2**.

**Step 5a:** If the yield is greater than the previous yield or the stochastic element meets the requirement of acceptance, continue by moving another step in the same direction.

**Step 5b:** If the yield is not greater than the previous one, then calculate the objective increment ( $\Delta y$ ) and test the element as follows: Randomly generate a random variable,  $x \sim \text{Uniform}(0, 1)$ . Apply the following rules then go to **Step 5a**.

**FFA element:** If  $x < P(\Delta y) \equiv \beta_0 \text{EXP}[\gamma(\Delta y)^2]$

**ACO element:** If  $x < P(\Delta y) \equiv \frac{\tau_i^\zeta \eta^\delta}{\tau_i^\zeta \eta^\delta + \tau_{i-1}^\zeta \eta^\delta}$

$$\tau_i = \rho \tau_{i-1} + \Delta \tau$$

where  $i$  is the number of iteration,  $\rho \tau_{i-1}$  is the concentration of the pheromone at the previous iteration and  $\Delta \tau$  is the cumulative process yields of the previous and current iterations.

**Step 5c:** Otherwise return to the preceding design point then carry out another  $2^3$  factorial design and return to **Step 2**. If the first step leads to a yield less than the yields obtained in the preceding  $2^3$  designs then replicate the design and go to **Step 2**.

**Step 6:** Implement the finishing strategy. This is a central composite design (CCD) centred on the point  $(T_{(in)p}, F_{(in)p})$ ,

$C_{A(in)p}$ ), and fit then a quadratic surface to find the maximum  $(T_{(in)p}, F_{(in)p}, C_{A(in)p})$ . If  $(T_{(in)p}, F_{(in)p}, C_{A(in)p})$  is within the volume of the designs, then  $(T_{(in)p}, F_{(in)p}, C_{A(in)p})$  is taken as the optimum operating condition. If  $(T_{(in)p}, F_{(in)p}, C_{A(in)p})$  is not within this volume, another CCD is carried out, centered on the point from the first CCD with the greatest yield. A quadratic surface is now fitted to all the data. If the maximum is outside the volume of the union of the two containing cubes, the ridge is searched for the greatest value of the function, using a step length of 0.05.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this work, the comparison is made with the measurement noise on the concentration of the desired product B (normal and independent with zero mean and standard deviation of 0.5, 1, 2 and 3). The typical three-dimensional response surfaces, with  $C_{A(in)}$  fixed at 1 and 15, are shown in Fig. 5. There are four performance measures over 100 realisations in this study. The first and second measures are an average ( $Y_{F1}$ ) and a standard deviation ( $Y_{F2}$ ) of greatest actual concentration of the desired product B from the finishing strategy respectively. The third is an average number of runs until the algorithms converge ( $Y_{F3}$ ). Finally it is the percentage of sequences ended at the optimum ( $Y_{F4}$ ).

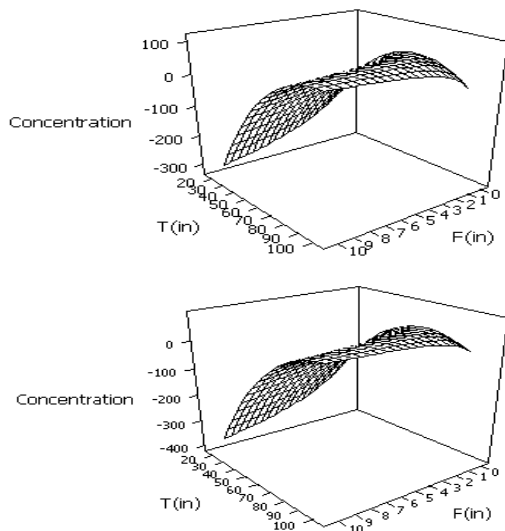


Fig. 5. The surface plot with  $C_{A(in)}$  fixed at 1 and 15 respectively.

The process settings for all the scenarios are given in Table II. The experimental results suggested that the conventional algorithm of SAM alone can produce an acceptable solution or even an optimal solution if the problem was not so noisy. When the problem is noisier, the integrated algorithms of SAM with FFA (SAMFFA) and SAM with ACO (SAMACO) are more suitable to exploit a solution space as a local search by embedding within the SAM. The exploitation process can be performed on each population member to improve its experience and thus obtain a population of local optimum solutions. However, the SAMACO requires higher levels of design points towards the optimum.

In Fig. 6, the actual yield achieved on one replicate of the

sequential procedures revealed the higher performance of both SAMFFA and SAMACO when compared with the SAM. Accordingly to noises, some factorial design points may need to duplicate and measure process yields. The performance of the method of steepest ascent and the integrated approach can be explained by the box plots in Fig. 7 when the error standard deviation was 2.0 and 3.0. Note that since the efficiency of these algorithms is related to their initial design points based on factorial designs, it would be helpful to set random starting design points for all algorithms. These results show that the performance of the integrated approach under the stochastic element of FFA and ACO seems superior to the conventional method of steepest ascent at the higher levels of error standard deviations. Moreover, percentage of sequences ended at the optimum or near optimum of radius equalling two from the integrated approach is better at higher levels of error standard deviation although the greater number of runs were required to converge to the optimum.

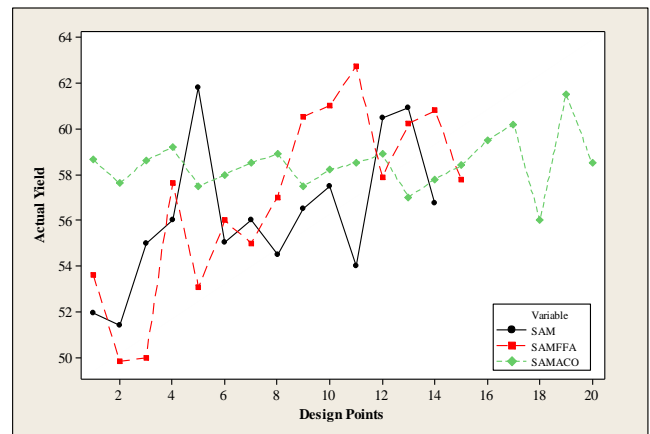


Fig. 6. Sequential performance of all algorithms on the CSTR with noise standard deviation of 2.0.

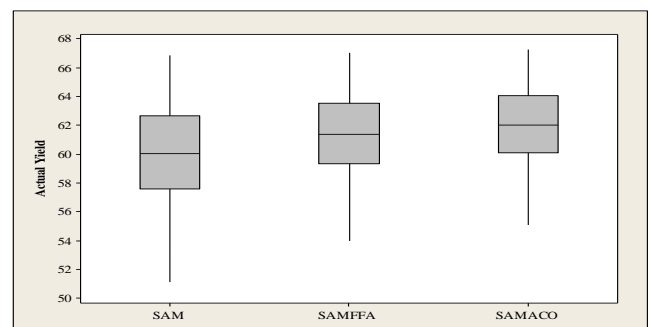
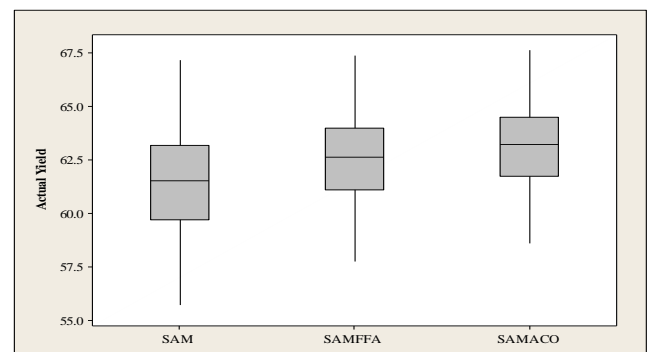


Fig. 7. Performances of all algorithms when the error standard deviation was 2.0 and 3.0 respectively.

TABLE II: FOUR ACHIEVEMENTS OVER 100 REALISATIONS

Performance Measures	Standard Deviation of Noise			
	0.5	1.0	2.0	3.0
$Y_{F1}$	59.3025 <sup>A</sup>	57.2845	59.3670	60.0308
	56.7503 <sup>B</sup>	57.6083	61.6109	61.6276
	57.7124 <sup>C</sup>	57.9102	61.5745	61.0987
$Y_{F2}$	7.6283	7.3860	9.5658	7.7273
	10.2718	6.8989	7.6716	6.4331
	11.3545	7.0215	7.8541	6.1568
$Y_{F3}$	33.03	33.57	32.06	31.53
	38.17	34.90	34.75	33.30
	40.25	35.69	36.25	35.95
$Y_{F4}$	0.85	0.87	0.82	0.78
	0.85	0.90	0.90	0.92
	0.83	0.89	0.89	0.85

Note: SAM<sup>A</sup>, SAMFFA<sup>B</sup>, SAMACO<sup>C</sup>

To find out the best solution within a limited time, is obviously difficult. There are various modifications to overcome in the research. However, convergence to the global optimum is not substantially rapid. Recommendations should be made for the values of the algorithm parameters, although these values depend on the selected performance measure. As stated earlier, the system in this research was restricted to three predictor variables. Consequently, comparisons and conclusions among the three algorithms may not be valid for other families of systems. Other stochastic approaches could be extended to the method based on conventional factorial designs to increase its performance, especially in terms of speed of convergence, when the error standard deviation is at higher levels.

#### REFERENCES

- [1] P. Luangpaiboon, "Noisy response surface optimisations via harmony search algorithm and its hybridisations," *Journal of Applied Operational Research*, vol. 3, no. 2, pp. 46-57, 2011.
- [2] E. Emad, H. Tarek, and G. Donald, "Comparison among five evolutionary-based optimisation algorithms," *Advanced Engineering Informatics*, vol. 19, pp. 45-52, 2005.
- [3] S. Lukasik and S. Zak, "Firefly algorithm for continuous constrained optimisation tasks," *Lecture Notes in Computer Science*, vol. 5796, pp. 97-106, 2009.
- [4] S. H äckel and P. Dippold, "The bee colony-inspired algorithm (BCiA): A two-stage approach for solving the vehicle routing problem with time

- windows," in *Proc. Genetic and Evolutionary Computation Conference*, Montreal, 2009, pp. 25-32.
- [5] K. S. Lee and Z. W. Geem, "A new metaheuristic algorithm for continuous engineering optimisation: Harmony search theory and practice," *Comput. Meth. Appl. Mech. Eng.*, vol. 194, pp. 3902-3933, 2004.
- [6] I. D. R. Muller, "Issues in bayesian analysis of neural network models," *Neural Computation*, vol. 10, pp. 571-592, 1995.
- [7] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: optimisation by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics Part B*, vol. 26, no. 1, pp. 29-41, 1996.
- [8] J. Y. Jeon, J. H. Kim, and K. Koh, "Experimental evolutionary programming-based high-precision control," *IEEE Control Sys. Tech.*, vol. 17, pp. 66-74, 1997.
- [9] R. Storn, "System design by constraint adaptation and differential evolution," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 1, pp. 22-34, 1999.
- [10] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 58-73, 2002.
- [11] A. Lokketangen, K. Jorsten, and S. Storoy, "Tabu search within a pivot and complement framework," *International Transactions in Operations Research*, vol. 3, pp. 305-316, 1994.
- [12] V. Granville, M. Krivanek, and J. P. Rasson, "Simulated annealing: a proof of convergence," *Pattern Analysis and Machine Intelligence, IEEE Transactions*, vol. 16, no. 6, pp. 652-656, 1994.



**P. Luangpaiboon** has been a lecturer, and Associate Professor, in the Industrial Statistics and Operational Research Unit (ISO-RU), the department of Industrial Engineering at Thammasat University, THAILAND since 1995. He graduated his Bachelor (1989-1993) and Master Degrees (1993-1995) in Industrial Engineering from Kasetsart University, THAILAND and Ph. D. (1997-2000) in Engineering Mathematics from Newcastle upon Tyne, ENGLAND. His research interests consist of meta-heuristics, optimisation, industrial statistics, the design and analysis of experiments and response surface methodology. He received Kasetsart University Master Thesis Award in 1995 (Dynamic Process Layout Planning), Certificate of Merit for The 2009 IAENG International Conference on Operations Research (A Hybrid of Modified Simplex and Steepest Ascent Methods with Signal to Noise Ratio for Optimal Parameter Settings of ACO), Best Paper Award for the Operations Research Network Conference 2010 (An Exploration of Bees Parameter Settings via Modified Simplex and Conventional Design of Experiments), Certificate of Merit for The 2011 IAENG International Conference on Operations Research (Bees and Firefly Algorithms for Noisy Non-Linear Optimisation Problems) and Best Student Paper Award for The 2011 IAENG International Conference on Industrial Engineering (Simulated Manufacturing Process Improvement via Particle Swarm Optimisation and Firefly Algorithms). His email address is lpongch@enr.tu.ac.th.