Optimal Cost Driver Selection in Activity-Based Costing Using Shuffled Frog Leaping Algorithm

Rungchat Chompu-inwai
Department of Industrial Engineering
Faculty of Engineering, Chiang Mai University
Chiang Mai, 50200, Thailand
rungchatc@hotmail.com

Trasapong Thaiupathump
Department of Computer Engineering
Faculty of Engineering, Chiang Mai University
Chiang Mai, 50200, Thailand
trasapong.t@cmu.ac.th

Abstract

Activity-based costing (ABC) system more precisely allocates the overhead costs to cost objects (products, services or customers) than traditional costing systems. In ABC, resources are consumed by the activities, and multiple cost drivers are used to allocate the costs of activities to the products. The selection of activity and cost drivers is then highly significant. Using too few cost drivers may result in low level of accuracy in allocating the overhead costs. On the other hands, a high accuracy normally requires a large number of cost drivers which would be very time-consuming and expensive in data collection, processing, and reporting. Therefore, the trade-off between the product cost accuracy and the ABC complexity is crucial. Using appropriate number of cost drivers is required to achieve a satisfactory level of information cost and accuracy, as well as to make the ABC system simpler to implement. The cost-drivers optimization (CDO) problem focuses on selecting the representative cost drivers by considering the trade-off between the information-gathering costs and the benefits of precise costing. Recently, many approaches have been applied to solve the CDO problem. In this paper, Shuffled Frog Leaping Algorithm (SFLA), a meta-heuristic method for finding optimal solutions, is applied in selecting optimal representative cost drivers. The objective function of the algorithm is the cost saving from the information gathering cost of eliminated cost drivers minus the loss of accuracy cost. With computational results, SFLA can effectively find the optimal cost driver combination that has the optimal objective function value. Convergence performances of the best and average objective function value are presented.

Keywords
Cost-drivers optimization, activity-based costing, shuffled frog leaping algorithm.

1. Introduction

While direct material costs and direct labor costs can be assigned directly to the product costs, the overhead costs (i.e. indirect material costs, direct labor costs, etc.) cannot be assigned to the products directly. In the traditional costing methods, product quantity or volume is mainly used in allocating the overhead costs to the products. Based on this method, the higher overhead costs are allocated to the higher production quantity products which may not reflect the resources used during manufacturing. On the other hands, based on the Activity-Based Costing (ABC) system, activities are performed to produce products and these activities consume resources. Therefore, the overhead costs are allocated to products based on activities performed using "Cost Drivers" that related to activities (Cooper and Kaplan, 1991, 1992). Cost drivers are factors that effect, cause or drive costs of activities performed (Barfield, Raibom, and

Kinney, 1994; Schniederjans and Garvin, 1997). Based on ABC, the higher the activities performed in production, the higher the overhead costs allocated to the product. To conclude, ABC method enhances accuracy in cost allocation by using multiple cost drivers in allocating the activity costs to the products, based on the resources used by those activities (Babad and Balachandran, 1993).

Although ABC method can achieve a more accurate cost assignment to products, a lot of time, employee and money has to be invested in implementing the system. In general, the more information needed, the greater the cost. It was reported by Kaplan and Anderson (2004) that, in brokerage operation of a large bank, the ABC data-gathering process required more than 70,000 employees at hundreds of bank facilities to record their time allocation in monthly reports. Fourteen full-time employees were employed to manage the process of data collection, data processing, and reporting. Moreover, as the number of involved activities increases, the demands on the computer facilities required to effectively store and process the data increase as well. The generic spreadsheet tools and other ABC software packages are not capable of handling the enormous data size and complexity. One month's data could require several days to process. For example, Hendee Enterprises, a $12 million fabricator of awnings, having 40 departments, 150 activities, 10,000 orders, and 45,000 line items, required three days to calculate ABC costs. In addition, Everaert, Bruggeman and Creus (2008) reported the difficulties of the ABC implementation in their proposed instructional case based on the experiences in a company named Sanac, a family-run plant-care product distributor in Belgium. Besides the fact that the data collection in ABC system would be very time-consuming, selecting the appropriate activity cost driver for many activities was difficult since the activities depended on several drivers. Using too many cost drivers can cause in excessive details that limit the acceptance and usefulness of the ABC system (Barfield, Raibom, and Kinney, 1994; Schniederjans and Garvin, 1997). Therefore, the selection of activity and cost drivers is greatly significant (Schniederjans and Garvin, 1997).

In designing an ABC system, two separate but interrelated decisions must be made: the number of cost drivers required and the selection of cost drivers to be used (Balad and Balachadran, 1993). With ABC, the more cost drivers the costing system uses, the more accurate the costs, but the more expensive data-gathering costs (Levitan and Gupta, 1996). Using too few cost drivers may result in low levels of accuracy in allocating the overhead costs. On the other hands, a high accuracy often requires a large number of cost drivers which would be very time-consuming and expensive in data collection, processing, and reporting. Therefore, the trade-off between the accuracy of the product cost and the complexity of ABC is crucial (Homburg, 2001). Using an appropriate number of cost drivers is needed to achieve a satisfactory level of information cost and accuracy, as well as to make the ABC system simpler to implement. Finding such an optimal trade-off is a challenging task. Balad and Balachadran (1993) developed a cost-drivers optimization (CDO) model and proposed greedy algorithms to find representative cost drivers by considering the trade-off between the information-gathering costs and the benefits of accurate costing information. Various interesting techniques have been applied to find optimal cost drivers, such as genetic algorithm (Levitan and Gupta, 1996), neural network (Urkmez *et al*., 2008), genetic algorithm and neural network hybrid (Kim and Han, 2003). Ramadan (2015) optimized the selection with specified cost budget using quasi-knapsack structure. Wang *et al*. (2010) applied the linear regression analysis and the maximum r-square improvement model selection method to the problem.

Shuffled frog leaping algorithm (SFLA), developed by Eusuff and Lansey (2003), is a memetic meta-heuristic approach to solve combinatorial optimization problems. SFLA has been known for a fast and efficient optimization method. The purpose of this paper is to apply the SFLA for the cost-drivers optimization (CDO) problem. SFLA is applied to two problems adapted from Babad and Balachandran (2003) and Smith and Leksan (1991) (see Tables 2 and 3). Results and performances of SFLA-based approach are illustrated.

2. Shuffled Frog Leaping Algorithm (SFLA)

SFLA is a meta-heuristic optimization technique based on the behavior of a frog colony in searching for the location having the maximum food. SFLA, originally proposed by Eusuff and Lansey in 2003, is capable in efficiently solving many complex optimization problems. SFLA combines the advantages of both the genetic-based memetic algorithm and the social behavior-based particle swarm optimization algorithm.

SFLA is a population based random search method. A population of possible solutions or virtual frogs in SFLA is divided into several groups referred to as memeplexes. Local search is performed by each frog in the memeplexes. Within each memeplex, the behavior of each frog can be affected by other frogs' behaviors, and frogs evolve through a memetic evolution process. After a number of predefined steps of memetics evolution, the memeplexes are mixed together and sorted in descending order of frogs' performance or fitness values. New memeplexes are then created by a shuffling process. The local search process and the shuffling process carry on until terminating criteria are met. A detailed pseudocode for SFLA is shown in Fig.1. Key parameters of SFLA are: number of frogs in each memeplex (n), number of memeplexes (m), number of generation for each memeplex before shuffling (N), number of shuffling iterations or generations, and the maximum leaping step size.

```
Initialize random positions of F frogs (solutions)
(a solution is a vector of group numbers of all cost drivers)
Do
    Sort the F frogs in descending order of performance or fitness values.
    Let Xg be the position of the global best performance frog.
    Divide the population into m memeplexes, each has n frogs;
    Frog i is assigned to memeplex r = i mod m
    For each memeplex:
        Construct a submemeplex of q frogs selected randomly according to the probability
            pj = 2(n+1–j)/n(n+1), j= 1, … , n.
        For the submemeplex;
            Let Xb and Xw be the position of the best and worst frog in the submemeplex;
            Calculate the new position of the worst frog (Xnew1) by
                Xnew1 = Xw + r (Xb – Xw)   ; where r is a uniform random number between 0 and 1
                (Moving toward submemeplex best frog)
            If Xnew1 performance is worse than Xw performance
                Calculate the new position of the worst frog (Xnew2) by
                    Xnew2 = Xw + r (Xg – Xw)   ; where r is a uniform random number between 0 and 1
                    (Moving toward global best frog)
                If Xnew2 performance is worse than Xw performance
                    Randomly generate a new frog to replace Xw
                else
                    Replace Xw with Xnew2
            else
                Replace Xw with Xnew1
            Sort the n frogs in memeplex in descending order
            Repeat for N iterations
    All the memeplexes are mixed together
While (terminating conditions are not satisfied)
```

Figure 1. SFLA pseudocode

2.1 Objective Function

The objective function for SFLA for cost drivers optimization problem is defined as the maximization of cost saving minus the cost of accuracy loss, where the cost saving is the total cost of all the drivers eliminated in the combinations and the cost of accuracy loss is defined as the square root of sum of squared difference between the product cost as computed using all cost drivers and the product cost using only the selected cost drivers. The objective function can be represented by the following equation:

$$\sum_{j} S_j C_j - \sqrt{\sum\nolimits_{i} (U_i - U_i')^2} \tag{1}$$

where

C_j = cost related with the cost driver j

$S_j = \begin{cases} 1 \text{ when cost driver } j \text{ is selected,} \\ 0 \text{ otherwise} \end{cases}$

U_i = total cost of product i obtained by using all cost drivers

$$U_i = \sum_{j} D_j V_{ij} \tag{2}$$

D_j = total cost of activities associated with cost driver j

V_{ij} = relative frequency of the usage of cost driver j by product i

$$V_{ij} = \frac{\bar{V}_{ij}}{\sum_i \bar{V}_{ij}} \tag{3}$$

$\bar{V}_{ij}$ = usage volume of cost driver j by product i
U_i' = total cost of product i obtained by using only the selected cost drivers

Assume that $\{m_1, m_2, \ldots, m_L\}$ are the selected cost drivers, and L is the total number of selected cost drivers. The total cost of product i computed using only the selected cost drivers is defined as

$$U_i' = \sum_{l=1}^{L}\left(V_{im_l}\left(\sum_{j\in G_l} D_j\right)\right) \tag{4}$$

V_{im_l} = relative frequency of the usage of cost driver m_l by product i
G_l = group of cost drivers that are represented by cost driver m_l

2.2 Representation

Proper representation of the solutions (virtual frogs) is vital for the success of the SFLA approach. In the CDO problem considered in this paper, the virtual frog is represented by a string of integer numbers, with the position of a number in the solution string representing the cost driver number and the length of the solution string represents the number of drivers considered in the problem. Combining the costs associated with several drivers into a group, and using only one representative driver from the group to drive the cost of the entire group result in saving information gathering costs. The main goal is to find the combination of representative cost drivers that can save reasonable amount of information gathering cost and loss of accuracy cost.

In a solution string, each number represent a group number, which is an integer value between 1 and the maximum number of groups allowed. For example, the solution string

1 1 2 4 2 3 1 1 2 4

represents a 10 cost drivers partitioned into 4 groups. The solution string can be interpreted as following: drivers 1, 2, 7 and 8 are in group 1, drivers 3, 5 and 9 are in group 2, only driver 6 is in group 3, and drivers 4 and 10 are in group 4. For each group, there is an optimal driver into which the rest of the drivers in the group is combined. The optimal driver is determined by calculating total savings (i.e. information cost saving minus loss of accuracy cost) for each driver in that group, and then selecting the driver which gives maximum total savings. For example, if the combination of cost driver 7 from group 1, cost driver 5 from group 2, cost driver 6 from group 3 and cost driver 10 from group 4 is the maximum saving, the parameters in Eq. (4) are as followings: L = 4, selected cost driver $\{m_1, m_2, \ldots, m_4\}$={7, 5, 6, 10}. The group of cost drivers that are represented by cost driver m_1= 7 is G_1= {1, 2, 7, 8}. While the group of cost drivers G_2= {3, 5, 9}, G_3= {6} and G_4= {4, 10} are represented by cost drivers m_2=5, m_3=6, m_4=10 respectively.

2.3 Example of frog leaping

According to the leaping rule of SFLA, $\boldsymbol{X}_{\text{new}} = \boldsymbol{X}_{\text{w}} + r\,(\boldsymbol{X}_{\text{b}} - \boldsymbol{X}_{\text{w}})$, a frog at $\boldsymbol{X}_{\text{w}}$ position attempts to leap toward the position of a better frog $\boldsymbol{X}_{\text{b}}$. If $\boldsymbol{X}_{\text{w}}$ = [1 1 2 4 2 3 1 1 2 4], $\boldsymbol{X}_{\text{b}}$ = [4 4 3 4 2 2 1 3 2 1] and random number r = 0.6, the new position of a worse frog is calculated as follows

$$\begin{aligned}\boldsymbol{X}_{\text{new}} &= [1\ 1\ 2\ 4\ 2\ 3\ 1\ 1\ 2\ 4] + 0.6\ ([4\ 4\ 3\ 4\ 2\ 2\ 1\ 3\ 2\ 1] - [1\ 1\ 2\ 4\ 2\ 3\ 1\ 1\ 2\ 4]) \\ &= [2.8\ 2.8\ 2.6\ 4\ 2\ 2.4\ 1\ 2.2\ 2\ 2.2]\end{aligned}$$

Since the numbers in solution vector $\boldsymbol{X}_{\text{new}}$ are needed to be rounded to the nearest integer,

$$\boldsymbol{X}_{\text{new}} = [3\ 3\ 3\ 4\ 2\ 2\ 1\ 2\ 2\ 2]$$

This new solution string $\boldsymbol{X}_{\text{new}}$ can be interpreted as follows: driver 6 is in group 1, drivers 5, 6, 8, 9, 10 are in group 2, drivers 1, 2, 3 are in group 3 and driver 4 is in group 4. In some cases, there are some benefits to have extra constraints on the maximum and minimum value of the random number *r*, since having *r* too close to 1 might cause the frog to be moved to the position that is too close to the position of the best frog. This might result in the algorithm converging on undesirable local minima. On the other hand, too small *r* might cause a slow convergence rate.

2.4 Simulation Parameter Settings

SFLA simulations are conducted with the algorithm parameter settings according to Table 1.

Table 1. SFLA parameter settings

SFLA parameter	Value
Number of frogs in a memeplex (n)	10
Number of memeplexes (m)	5
Total frog population ($F = n \times m$)	50
Number of frogs in a submemeplex (q)	7
Number of evolutionary steps in each generation (N)	10
Number of generations	16

2.5 Problems

Problem 1

Problem 1 sample data, adapted from Babad and Balachandran (1993), are shown in Table 2. In this problem, there are four products (Product 1-Product 4). The ideal product costs using all drivers are $1,026 for Product 1, $2,550 for Product 2, $1,338 for Product 3 and $5,110 for Product 4.

Table 2. Product Costing Data for Problem 1 (adapted from Babad and Balachanran, 1993)

Product	Quantity	Material (Dollars)	Direct Labor	Machine Hours	Setups	Orders	Time Handled	Parts	Resulting Product Cost and Objective Function Values	
		(D1)	(D2)	(D3)	(D4)	(D5)	(D6)	(D7)	Ideal	SFLA
Product 1	10	60	5	10	1	1	1	1	1026	1253
Product 2	100	600	50	40	2	3	3	1	2550	2506
Product 3	10	180	15	20	1	1	1	1	1338	1253
Product 4	100	1800	150	100	4	3	3	1	5110	5012
Unit Consumed		2640	220	170	8	8	8	4		
Dollar Value		264	2200	3400	960	1000	200	2000		
Information Cost			2500	1500	2000	2000		2500	10500	2000
Cost Savings									0	8500
Loss of Information									0	265.17
Objective Function Value									0	8234.83

Problem 2

Problem 2 sample data, adapted from a case study by Smith and Leksan (1991), are shown in Table 3. This problem has three product (C, M, L) and ten cost drivers. The ideal solution using all ten drivers assigns $12,340 to product C, $8,320 to Product M and $4,340 to Product L.

Table 3. Product Cost Data for Problem 2 (adapted from Smith and Leksan, 1991)

Product	Quantity	Purchasing Materials	Purchasing Components	Number of Venders	Number of Units	Units Expedited	Part Shipments	Materials Shipments	Production Runs	Total Shipments	Inspection Points	Resulting Product Cost and Objective Function Values	
		(D1)	(D2)	(D3)	(D4)	(D5)	(D6)	(D7)	(D8)	(D9)	(D10)	Ideal	SFLA
Product C	7000	500	2000	25	7000	1500	2000	500	10	2500	196000	12350	12500
Product M	2900	300	1500	30	2900	450	1500	300	15	1800	118900	8321.25	8250
Product L	100	200	500	45	100	50	500	200	25	700	5100	4328.75	4250
Unit Consumed		1000	4000	100	10000	2000	4000	1000	50	5000	320000		
Dollar value		2000	1000	2000	2000	1000	5000	2500	2500	3000	4000		
Information Cost		1500	1800	1700	1800	2000	1800	2200	2300	2400	1700	19200	3300
Cost Savings												0	15900
Loss of Information												0	183.79
Objective Function Value												0	15716.21

3. Numerical Results

When the SFLA was applied to the Problem 1 costing data set, the results suggested that only one driver, Setups, should remain. All overhead costs should be driven by Setups. This solution produced a substantial savings in information-gathering costs of $8500. Using Setups as the only driver would allocate $1,253 to Product 1, $2,506 to Product 2, $1,253 to Product 3 and $5,012 to Product 4. The costs per unit for Product 1-4 were $125.30, $25.06, $125.30 and $50.12, compared with the ideal unit cost using all drivers for Product 1-4 of $102.60, $25.50, $133.80 and $51.10. With the objective function defined in Eq.(1), the best objective function value was $8,234.83 which consisted of $8,500 cost savings and $265.17 loss of accuracy cost.

For Problem 2 costing data set, this problem initially had ten cost drivers. The results of SFLA suggested that only two drivers - ‘D1-Purchasing Materials’ and ‘D6-Part Shipments’ - were necessary. These two cost drivers represented two cost driver pools. The cost drivers ‘D1-Purchasing Materials’, ‘D2-Purchasing Components’, ‘D4-Number of Units’, ‘D5-Unit Expedited’, ‘D7-Materials Shipments’, ‘D8-Production Runs’ and ‘D10-Inspection Points’ were represented by cost driver ‘D1-Purchasing Materials’. The cost drivers ‘D3-Number of Vendors’, ‘D6-Part Shipments’ and ‘D9-Total Shipments’ were represented by cost driver ‘D6-Part Shipments’. SFLA allocated $12,500 to Product C, $8,250 to Product M and $4,250 to Product L. The costs per unit for Product C, Product M, and Product L were $1.76, $2.84 and $42.50, compared with the ideal unit cost using all drivers for Product C, Product M, and Product L of $1.76, $2.86 and $43.29. SFLA achieved the best objective function value of $15,716.21 which consisted of the $15,900 cost savings and $183.79 accuracy loss cost.

With 20 repeat trials with different initial random frog population and algorithm parameter settings according to Table 1, Figs.2 and 3 show the convergence performance for Problem 1 and 2. The best and average objective function value are shown with dashed line and solid line respectively. In 20 experiment trials, there were 17 results (85%) for Problem 1 and 16 results (80%) for Problem 2 having the average objective function values of frog population converged to within 10% of the best possible objective function value. The best frog could achieve a near optimal solution within the first five generations on average for both Problem 1 and 2.

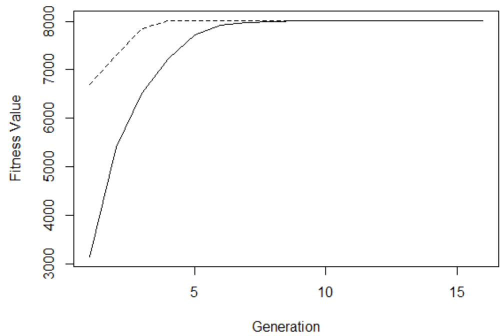

Figure 2. Average fitness and best fitness performance of SFLA for Problem 1

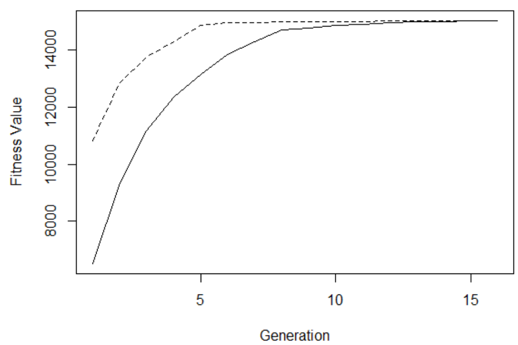

Figure 3. Average fitness and best fitness performance of SFLA for Problem 2

4. Conclusions

This paper presents an SFLA approach to finding the optimal combination of cost drivers which balance the trade-off between the information gathering cost saving and the loss of accuracy cost. Majority of the experimental results show that the frog population in SFLA efficiently converges to proximity of the optimal solutions. However, there are few results showing the algorithm still converges to undesirable local optima. Convergence performance depends largely on the selection of the algorithm parameters. This proposed method may be applied as one of the solutions to the difficulties with ABC that were discussed earlier, such as time consumed in data collection, processing, and reporting, as well as the difficulty in selecting the appropriate activity cost drivers.

References

Babad, Y. M., and Balachandran, B. V., Cost driver optimization in activity-based costing, *The Accounting Review*, vol. 68, no. 3, pp. 563-575, 1993.

Barfield, J.T., Raibom, C.A., and Kinney, M.R., *Cost Accounting: Traditions and Innovations*, 2nd ed., West, Minneapolis/ St.Paul, MN, 1994.

Cooper, R., and Kaplan, R. S., Profit priorities from activity-based costing, *Harvard Business Review*, vol. 69, no. 3, pp. 130–136, 1991.

Cooper, R. and Kaplan, R. S., Activity-based costing: Measuring costs of resource usage, *Accounting Horizons*, vol. 6, no. 3, pp. 1–13, 1992.

Eusuff, M., Lansey, K, and Pasha, F., Shuffled frog leaping algorithm: a memetic meta-heuristic for discrete optimization, *Engineering Optimization*, vol. 8, no. 2, pp. 129-154, 2006.

Everaert, P., Bruggeman, W. and Creus, G.D., Sanac Inc.: From ABC to time-driven ABC (TDABC) – An instructional case, *Journal of Accounting Education*, vol. 26, no. 3, pp. 118-154, 2008.

Homburg, C., A note on optimal cost driver selection in ABC, *Management Accounting Research*, vol. 12, pp. 197-205, 2001.

Kaplan, R. S., and Anderson, S. R., Time-Driven Activity-Based Costing, *Harvard Business Review*, vol. 82, no. 11, pp. 1–8, 2004.

Kim, K. J., and Han, I., Application of a hybrid genetic algorithm and neural network approach in activity-based costing, *Expert Systems with Applications*, vol. 24, pp. 73-77, 2003.

Levitan, A., and Gupta, M., Using genetic algorithms to optimize the selection of cost drivers in activity-based costing, *Intelligent Systems in Accounting, Finance and Management*, vol. 5, pp. 129-145, 1996.

Ramadan, S. Z., Optimizing the selection of cost drivers in activity-based costing using quasi-knapsack structure, *International Journal of Business and Management,* vol. 10, no. 7, pp.74-84, 2015.

Schniederjans, M. J., and Garvin, T., Using the analytic hierarchy process and multi-objective programming for the selection of cost drivers in activity-based costing, *European Journal of Operational Research*, vol. 100, no. 1, pp. 72-80, 1997.

Smith, K. V., and Leksan, M. P., A manufacturing case study on activity-based costing, *Journal of Cost Management*, vol. 5, no. 2, pp. 45-54, 1991.

Urkmez, S., Bilgili, E., Ziarati, R., and Stockton, D., Application of novel artificial intelligent techniques in shipbuilding using activity based costing and neural networks, *Proceedings of 2008 International Maritime Lecturers Association (IMLA 2008)*, Izmir, 2008.

Wang, P., Du, F., Lei, D., and Lin, T., The choice of cost drivers in activity-based costing: application at a Chinese oil well cementing company. *International Journal of Management*, vol. 27, no. 2, pp. 367-380, 2010.

Biography

Rungchat Chompu-inwai is currently an assistant professor in Industrial Engineering at Chiang Mai University, Thailand. She received her B.Eng. degree in Industrial Engineering from Chiang Mai University, Thailand in 1997, M.Eng.Sc. degree in Manufacturing Engineering from The University of New South Wales, Sydney, Australia in 2000, and Ph.D. in Industrial and Manufacturing Engineering from Oregon State University, USA in 2006. Her teaching and research interests include lean manufacturing, process and quality improvement, application of Industrial Engineering (IE) techniques for productivity improvement, economic analysis, and development of new innovative and effective pedagogic approaches for Engineering Education.

Trasapong Thaiupathump is currently an associate professor in Computer Engineering at Chiang Mai University, Thailand. He received his B.Eng. degree in Computer Engineering from King Mongkut's Institute of Technology, Ladkrabang, Thailand in 1993, M.S. degree in Computer Engineering from University of Southern California, USA in 1996, and Ph.D. degree in Electrical Engineering from University of Pennsylvania, USA in 2002. His research interests are computer communications and computer applications in manufacturing.