A Computer Store Optimization Model using Particle Swarm Optimization and Simulated Annealing

Aaron Don M. Africa
Department of Electronics and Communications Engineering, De La Salle University Manila, Philippines
Email: aaron.africa@dlsu.edu.ph

***Abstract*—In a computer store business, the number of units in the inventory and specification of the parts of the units play a big factor in its success. There will be a higher rate of success if there are more computer units to sell and advanced specifications for the computer parts but it will cost a lot of money. In maintaining a computer store, resources have to be optimized to its fullest capacity. Given a budget constraint, the specifications of the parts of the computers that will be purchased have to be optimized. This process is usually done manually by people who have a computer store which results in an inefficient allocation of resources. This research presents a scientific approach minimizing the average annual cost of ordering and storing the computer sets. This is done by using Simulated Annealing and Particle Swarm Optimization in purchasing the number of units and specification of the parts of the computers.**

***Index Terms*—simulates annealing, particle swarm optimization, optimization Techniques, and computer hardware**

I. Introduction

A computer store sells video cards, hard disk, network peripherals, whole computer units etc [1]. For this reason, the number of computer units and the specification of its parts play an important role in its profit and return of investment [2]. The more money you invest, you can buy more computers in the inventory with good specifications. The problem is the more money you invest in a computer store, the longer will be your Return of Investment (ROI) [3]. The depreciation value of the computer units will also have to be considered because after a few weeks, newer models are available so you have to sell them at the soonest possible time. In general, it is better to have a minimal capital but still have competitive computer units in the inventory [4].

This research will show a scientific approach in designing the specifications of the units in a computer store's inventory. It will take into consideration specifications like the video card memory, hard disk space, cost of the unit etc. Simulated Annealing and Particle Optimization (PSO) model would be used to minimize the average annual cost of ordering and storing the computer sets. A sample design will be presented taking into consideration factors like demand cost and ordering cost. The sample design will be simulated in a computer software that applies the Simulated Annealing and Particle Optimization (PSO).

II. Overview

A. Simulated Annealing

Discovering the optimal solution for certain optimization problems is a difficult task [5]. One reason is if a problem becomes large, we have to search on many optimal solutions to find the optimal one [6]. Even with advanced computing there are still many possible solutions to consider. In this case, we cannot find the optimal solution given a certain amount of time so we have to settle with the most optimal one [7]. One optimization algorithm that does this is the Simulated Annealing. Simulated Annealing is a technique to find a good solution to an optimization problem by trying to find random variations of the current solution [8]. Simulated Annealing makes use of principles derived from statistical mechanics for implementing global search [9].

A worse variation is accepted as the new solution with a probability that decreases as the computation proceeds. The slower the cooling schedule, or rate of decrease, the more likely the algorithm is to find an optimal or near-optimal solution [10]. Simulated Annealing was inspired by annealing in metal work. Annealing is the cooling and heating of materials with the goal of altering its physical structure due to changes in its internal structure [11]. In Simulated Annealing, a temperature variable is kept to simulate the heating process [12]. The temperature is initially set to high and it is allowed to cool down as the algorithm is run. The Simulated Annealing algorithm has the ability to jump at any local optimum. This algorithm has a feature called gradual cooling. This feature makes the algorithm effective in finding the close optimum solution. This is useful when dealing with problems that contain numerous local optimums [13].

B. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a swarm intelligence algorithm. It was developed by Dr. Eberhart and Dr. Kennedy in 1995 [14]. It was inspired by social

Manuscript received August 7, 2013; revised September 11, 2013.

doi: 10.12720/joace.2.1.20-25

behaviours like bird flocking. The PSO is initialized with a population of random solutions and searches for optima by updating generations. For example, there is a swarm of insects or a school of fish. The rest of the swarm will be able to follow quickly even if they are on the opposite side of the swarm [15]. The state space should be explored appropriately. For this to be done, each particle should have a level of randomness to their movement in order that the movement of the swarm has an explorative capability [16]. A particle of the swarm should be influenced by the rest of the swarm but it also has to explore independently to a certain extent. This behaviour is a manifestation of fundamental exploration-exploration problems that occurs in search problems [17]. The Particle Swarm Optimization (PSO) has been successfully applied to many research areas and applications over the years. It was shown that PSO has provided better and faster results compared to other methods [18].

III. Computer Store Sample Design

TABLE I. ORDERING COST AND DEMAND RATE

	Ordering cost *aj* (php)	Demand rate *dj*
1	55,000	800
2	35,000	400
3	40,000	1200
4	50,000	1000
5	40,000	500
6	35,000	800
7	25,000	500
8	40,000	600
9	50,000	700
10	55,000	800

TABLE II. COST, SPACE AND MEMORY SPECIFICATIONS

	Unit cost *cj* (php)	Computer sized memory *ej* (mb)	Hard Disk space *fj* (gb)	Memory of Video Card *ij* (mb)
1	50,000	256	80	128
2	30,000	512	120	64
3	45,000	256	40	64
4	55,000	256	200	256
5	50,000	256	80	64
6	40,000	512	120	128
7	30,000	1,000	160	128
8	35,000	512	80	64
9	55,000	256	200	128
10	60,000	512	120	64
Max	2,500,000	102,400	25,000	51,200

TABLE III. SPEED AND CACHE SPECIFICATIONS

	Network Support Speed *kj* (mbps)	Memory Speed *lj* (mhz)	Cache Memory *mj* (mb)
1	1,000	500	1
2	2,000	1,000	2
3	1,000	250	2
4	1,500	500	2
5	1,000	500	1
6	1,000	250	2
7	1,500	1,000	1
8	1,000	250	1
9	1,000	500	1
10	1,500	500	2
Max	2,000,000	75,000	120

This section shows a sample design of a computer store. The computer store stocks and sells ten different models of computer sets. The store cannot afford to have an inventory worth more than P2, 500,000 at any time. The computers are ordered in lots. It cost php a_j for the store whenever a lot of computer model *j* is ordered. The cost of one computer of model *j* is c_j. The demand rate of computer model *j* is d_j units per year. The rate at which the inventory costs accumulate is known to be proportional to the investment in the inventory at any time with $q_j = 0.5$, denoting the constant proportionality for computer model j. Each computer set occupies an area *of* $sj = 3.3\ m^2$ and the maximum storage space available is $250\ m^2$ .The processor speed of each computer is $n_j = 3\ Ghz$ with a maximum of 110 Ghz. The maximum number of computer model ordered in each lot is 20. The store has a strict compliance to store the maximum quantities of specifications.

The information is given below:

Tables I, II and III shows the computer specifications. Since the demand rate per year of model *j* is d_j , the number of times the computer model j needs to be ordered is d_j/x_j. The cost of ordering computer model j per year is thus a_jd_j / x_j , $j = 1,2,3\ldots10$. The cost of storing the computer sets of model *j* per year is $q_jc_jx_j / 2$ since the average level of inventory at any time during the year is equal to $c_jx_j/2$. Thus the objective function (cost of ordering plus storing) can be expressed as:

$$f(X) = (a_1d_1 / x_1 + q_1c_1x_1 / 2) + (a_2d_2 / x_2 + q_2c_2x_2 / 2) + (a_3d_3 / x_3 + q_3c_3x_3 / 2) + (a_4d_4 / x_4 + q_4c_4x_4 / 2) + (a_5d_5 / x_5 + q_5c_5x_5 / 2) + (a_6d_6 / x_6 + q_6c_6x_6 / 2) + (a_7d_7 / x_7 + q_7c_7x_7 / 2) + (a_8d_8 / x_8 + q_8c_8x_8 / 2) + (a_9d_9 / x_9 + q_9c_9x_9 / 2) + (a_{10}d_{10} / x_{10} + q_{10}c_{10}x_{10} / 2)$$

where the design vector ***X*** is given by:

$$X = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{Bmatrix}$$

The constraint on the volume of inventory can be stated as:

$$c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 + c_5x_5 + c_6x_6 + c_7x_7 + c_8x_8 + c_9x_9 + c_{10}x_{10} \le 2,500,000$$

The limitation of the storage area is given by:

$$s_1x_1 + s_2x_2 + s_3x_3 + s_4x_4 + s_5x_5 + s_6x_6 + s_7x_7 + s_8x_8 + s_9x_9 + s_{10}x_{10} \le 250$$

The limitation on the computer sized memory can be given by:

$e_1x_1 + e_2x_2 + e_3x_3 + e_4x_4 + e_5x_5 + e_6x_6 + e_7x_7 + e_8x_8 + e_9x_9 + e_{10}x_{10} \leq 102400$

The limitation on the Hard Disk Space is given by:

$f_1x_1 + f_2x_2 + f_3x_3 + f_4x_4 + f_5x_5 + f_6x_6 + f_7x_7 + f_8x_8 + f_9x_9 + f_{10}x_{10} \leq 25{,}000$

The limitation on the video card memory is:

$i_1x_1 + i_2x_2 + i_3x_3 + i_4x_4 + i_5x_5 + i_6x_6 + i_7x_7 + i_8x_8 + i_9x_9 + i_{10}x_{10} \leq 51200$

The limit on the Network support speed is:

$k_1x_1 + k_2x_2 + k_3x_3 + k_4x_4 + k_5x_5 + k_6x_6 + k_7x_7 + k_8x_8 + k_9x_9 + k_{10}x_{10} \leq 2000000$

The limit on the memory speed is given by:

$l_1x_1 + l_2x_2 + l_3x_3 + l_4x_4 + l_5x_5 + l_6x_6 + l_7x_7 + l_8x_8 + l_9x_9 + l_{10}x_{10} \leq 75{,}000$

The limit on the Cache Memory is given by:

$m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4 + m_5x_5 + m_6x_6 + m_7x_7 + m_8x_8 + m_9x_9 + m_{10}x_{10} \leq 120$

The limit on the processor speed is given by:

$n_1x_1 + n_2x_2 + n_3x_3 + n_4x_4 + n_5x_5 + n_6x_6 + n_7x_7 + n_8x_8 + n_9x_9 + n_{10}x_{10} \leq 110$

Since the design variables cannot be negative and has a maximum of 20 we have

$0 \leq xj \leq 20, j = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

IV. Data and Results

A. Model of the Computer Store Design

The equation of X that should be minimized has the equation of:

$f(X) = (4.4 \text{ x } 10^7 / x_1 + 12{,}500x_1) + (1.4 \text{ x } 10^7 / x_2 + 7{,}500x_2) + (4.8\text{x}10^7 / x_3 + 11{,}250x_3) + (5 \text{ x } 10^7 / x_4 + 13{,}750x_4) + (2 \text{ x } 10^7 / x_5 + 12{,}500\, x_5) + (2.8 \text{ x } 10^7 / x_6 + 10{,}000\, x_6) + (1.25 \text{ x } 10^7 / x_7 + 7{,}500x_7) + (2.4 \text{ x } 10^7 / x_8 + 8{,}750x_8) + (3.5 \text{ x } 10^7 / x_9 + 13{,}750\, x_9) + (4.4 \text{ x } 10^7 / x_{10} + 15{,}000x_{10})$

Subject to

$G1(\mathbf{X}): 50{,}000x_1 + 30{,}0000x_2 + 45{,}000x_3 + 55{,}000x_4 + 50{,}000x_5 + 40{,}000x_6 + 30{,}000x_7 + 35{,}000x_8 + 55{,}000x_9 + 60{,}000x_{10} \leq 2{,}500{,}000$

$G2(\mathbf{X}): 3.3\,(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 250$

$G3(\mathbf{X}): 256x_1 + 512x_2 + 256x_3 + 256x_4 + 256x_5 + 512x_6 + 1000x_7 + 512x_8 + 256x_9 + 512x_{10} \leq 102400$

$G4(\mathbf{X}): 80x_1 + 120x_2 + 40x_3 + 200x_4 + 80x_5 + 120x_6 + 160x_7 + 80x_8 + 200x_9 + 120x_{10} \leq 25{,}000$

$G5(\mathbf{X}): 128x_1 + 64x_2 + 64x_3 + 256x_4 + 64x_5 + 128x_6 + 128x_7 + 64x_8 + 128x_9 + 64x_{10} \leq 51200$

$G6(\mathbf{X}): 1000x_1 + 2000x_2 + 1000x_3 + 1500x_4 + 1000x_5 + 1000 + 1500x_7 + 1000x_8 + 1000x_9 + 1500x_{10} \leq 2000000$

$G7(\mathbf{X}): 500x_1 + 1000x_2 + 250x_3 + 500x_4 + 500x_5 + 250x_6 + 1000x_7 + 250x_8\, 500 + 500x_{10} \leq 75{,}000$

$G8(\mathbf{X}): x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + 2x_6 + x_7 + x_8 + x_9 + 2x_{10} \leq 120$

$G9(\mathbf{X}): 3(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}) \leq 110$

$G10(\mathbf{X}): 0 \leq x_1 \leq 20$
$G11(\mathbf{X}): 0 \leq x_2 \leq 20$
$G12(\mathbf{X}): 0 \leq x_3 \leq 20$
$G13(\mathbf{X}): 0 \leq x_4 \leq 20$
$G14(\mathbf{X}): 0 \leq x_5 \leq 20$
$G15(\mathbf{X}) = 0 \leq x_6 \leq 20$
$G16(\mathbf{X}) = 0 \leq x_7 \leq 20$
$G17(\mathbf{X}) = 0 \leq x_8 \leq 20$
$G18(\mathbf{X}) = 0 \leq x_9 \leq 20$
$G19(\mathbf{X}) = 0 \leq x_{10} \leq 20$

B. Test Using Simulated Annealing

The first part of the simulation is to use Simulated Annealing (SA). The first step is to place the equations in the SAdemo file as shown in Table IV. Three tests are performed with the temperatures of 1.88E-05, 3.05E-05 and .000305.

TABLE IV. PROBLEMS INPUTTED IN THE SA DEMO FILE

Fitness	**4.8E+08**			
Variable	**Parameter**	**Max**	**Min**	**Value**
x1	0.225	20	0	4.501
x2	0.738	20	0	14.769
x3	0.109	20	0	2.186
x4	0.935	20	0	18.701
x5	0.01	20	0	0.2
x6	1	20	0	20
x7	0.01	20	0	0.2
x8	1	20	0	20
x9	1	20	0	20
x10	1	20	0	20
Obj. fcn.	205797			
Const (<)	**LHS**	**RHS**	**Penalty**	
1	5198090.43	2500000	2698090.4	
2	265.8462242	250	15.846224	
3	50152.98873	10240	39912.989	
4	16408.22894	25000	0	
5	14167.39082	512000	0	
6	154779.5829	2000000	0	
7	57217.70002	75000	0	
8	196.2172279	120	76.217228	
9	361.6783856	110	251.67839	
			273801926	

The first test of the model is with the following parameters:

TABLE V. SIMULATION WITH TEMPERATURE 1.88E-19

Parameter	Value
Tstart	1000
Cooling	0.999
Step	1
Iteration	50,000
Temp	1.88E-05

TABLE VI. SIMULATION WITH TEMPERATURE 1.88E-19

Fitness	65934241			
Variable	Parameter	Max	Min	Value
x1	0.264	20	0	5.290
x2	0.475	20	0	9.504
x3	0.356	20	0	7.120
x4	0.219	20	0	4.384
x5	0.101	20	0	2.034
x6	0.188	20	0	3.772
x7	0.131	20	0	2.623
x8	0.569	20	0	11.381
x9	0.255	20	0	5.105
x10	0.462	20	0	9.245
Obj. fcn.	64347264.5			
Const (<)	LHS	RHS	Penalty	
1	2421294.757	2500000	0	
2	152.1670775	250	0	
3	26109.76585	10240	15869.766	
4	6801.77269	25000	0	
5	5786.034317	512000	0	
6	78094.08426	2000000	0	
7	30726.79162	75000	0	
8	94.48989987	120	0	
9	181.3882935	110	71.388294	
			1586976.6	

Fitness is: 65934241

The second test of the model is with the following parameters:

TABLE VII. PARAMETERS OF THE TEST WITH A TEMPERATURE OF 3.05E-05

Parameter	Value
Tstart	100
Cooling	0.9997
Step	1
Iteration	50,000
Temp	3.05E-05

TABLE VIII. SIMULATION WITH TEMPERATURE 3.05E-05

Fitness	62900396			
Variable	Parameter	Max	Min	Value
x1	0.258	20	0	5.165
x2	0.582	20	0	11.655
x3	0.343	20	0	6.874
x4	0.326	20	0	6.528
x5	0.112	20	0	6.528
x6	0.488	20	0	9.761
x7	0.299	20	0	5.997
x8	0.395	20	0	7.913
x9	0.232	20	0	4.657
x10	0.190	20	0	3.803
Obj. fcn.	60975851.01			
Const (<)	LHS	RHS	Penalty	
1	2401229.693	2500000	0	
2	185.3047171	250	0	
3	29485.45484	10240	19245.455	
4	7725.001431	25000	0	
5	7025.984879	512000	0	
6	84433.2844	2000000	0	
7	34995.59551	75000	0	
8	103.2367647	120	0	
9	193.8404775	110	83.840477	
			1924545.5	

Fitness is: 62900396

The second part of the model is with the following parameters:

TABLE IX. PARAMETERS OF THE TEST WITH A TEMPERATURE OF .000305

Parameter	Value
Tstart	1000
Cooling	0.9997
Step	1
Iteration	50,000
Temp	.000305

TABLE X. SIMULATION WITH TEMPERATURE 0.000305

Fitness	62123300			
Variable	Parameter	Max	Min	Value
x1	0.231	20	0	4.621
x2	1	20	0	20
x3	0.260	20	0	5.200
x4	0.436	20	0	8.722
x5	0.235	20	0	4.715
x6	0.367	20	0	7.359
x7	0.440	20	0	8.803
x8	0.161	20	0	3.226
x9	0.201	20	0	4.022
x10	0.268	20	0	5.372
Obj. fcn.	59727486.88			
Const (<)	LHS	RHS	Penalty	
1	2425604.512	2500000	0	
2	206.7410971	250	0	
3	34198.13619	10240	23958.136	
4	9098.402262	25000	0	
5	7873.135986	512000	0	
6	103492.8399	2000000	0	
7	46476.90803	75000	0	
8	118.6989152	120	0	
9	216.1315048	110	106.1315	
			2395813.6	

Fitness is: 62123300

With the three tests that we performed, the best and the lowest fitness obtained were 62123300. This fitness was obtained in the third test with a 0.000305 temperature.

C. Test Using Particle Swarm Optimization

The second part of the simulation is to use Particle Swarm Optimization (PSO). The model of the computer store design was inputted in the Excel program that performs the PSO algorithm. Initializing the variables the following information is obtained:

TABLE XI. COMPUTER STORE DESIGN INPUTTED IN THE PSO DEMO FILE

Fitness	**4.4E+13**			
Variable	**Parameter**	**Max**	**Min**	**Value**
x1	0.779	20	0	15.581
x2	1E-07	20	0	2E-05
x3	1	20	0	20
x4	1E-07	20	0	2E-05
x5	0.697	20	0	13.959
x6	0.329	20	0	6.592
x7	0.694	20	0	13.896
x8	1E-07	20	0	2E-05
x9	0.303	20	0	6.060
x10	1	20	0	20
Obj. fcn.	4.4E+13			
Const (<)	**LHS**	**RHS**	**Penalty**	
1	4620908.8	2500000	2120909	
2	231.09336	250	0	
3	41745.033	10240	31505.03	
4	9789.7681	25000	0	
5	8846.0085	512000	0	
6	113037	2000000	0	
7	48344.492	75000	0	
8	142.68107	120	22.68107	
9	288.26689	110	178.2669	

Table XI shows the computer store design inputted in the PSO demo file after initialization.

After running the PSO Program we get the solution of:

TABLE XII. SOLUTION AFTER APPLYING PSO

Fitness	**4799893.2**			
Variable	**Parameter**	**Max**	**Min**	**Value**
x1	0.225	20	0	4.501
x2	0.738	20	0	14.769
x3	0.109	20	0	2.186
x4	0.935	20	0	18.701
x5	0.01	20	0	0.2
x6	1	20	0	20
x7	0.01	20	0	0.2
x8	1	20	0	20
x9	1	20	0	20
x10	1	20	0	20
Obj. fcn.	205797013			
Const (<)	**LHS**	**RHS**	**Penalty**	
1	5198090.4	2500000	2698090	
2	265.84622	250	15.84622	
3	50152.989	10240	39912.99	
4	16408.229	25000	0	
5	14167.391	512000	0	
6	154779.58	2000000	0	
7	57217.7	75000	0	
8	196.21723	120	76.21723	
9	361.67839	110	251.6784	

Table XII shows the solution after applying the Particle Swarm Optimization (PSO). The value of the optimal solution is: 47959893.2.

V. ANALYSIS AND CONCLUSION

This paper showed an expert optimization approach in minimizing the average annual cost of ordering and storing the computer sets in a computer store. Simulated Annealing and Particle Swarm Optimization (PSO) was used in the simulation. A sample computer store design was used with different constraint and conditions to be followed. A total of 50,000 iterations were used for the Simulation model. This paper gave the objective functions and the optimal solutions for both Simulated Annealing and PSO. This paper gave the optimal solutions with the two methods that will otherwise be impossible with casual methods because it is a non-linear equation.

The methodology shown in this paper can also be tested in other stores that have an objective of minimizing the average annual cost of ordering and storing. The process shown is not only limited in a computer store but also to other shops that have inventory. The parameters can also be expanded and different requirements like additional constraints can be added to the model.

REFERENCES

[1] R. Villman. (August 2013). Villman Computer Page. [Online]. Available: http://www.villman.com/

[2] ULOP. (August 2013). Unified Lanshop Owners. [Online]. Available: http://www.ulop.com/

[3] Entrepreneur Business Exchange. (July 2013). Entrepreneur Business Discussions. [Online]. Available: http://www.entrepreneur.com.ph

[4] Straight Line Depreciation. (July 2013). Depreciation Discussions. [Online]. Available: http://www.straightlinedepreciation.org/

[5] P. Wightman, A. Fabregasy, and M. Labradorz, "An optimal solution to the MCDS problem for topology construction in wireless sensor networks," presented at the 2010 IEEE Latin-American Conference on Communications, Bogota, September 15-17, 2010.

[6] N. Piersma and W. Dikj, "A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search," *Mathematical and Computer Modeling*, vol. 24, no. 9, pp. 11-19, November 1996.

[7] R. Eglese, "Simulated annealing: A tool for operational research," *European Journal of Operational Research*, vol. 46, no. 3, pp. 271-281, November 1990.

[8] S. Mousavi and R. Moghaddam, "A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain," *Journal of Manufacturing Systems*, vol. 32, no. 2, pp. 335-347, April 2013.

[9] D. Foo, M. El-Halwagi, and R. Tan, *Recent Advances in Sustainable Process Design and Optimization*, World Scientific Publishing Company, 2012.

[10] C. Roa-Sepulveda and B. Pavez-Lazo, "A solution to the optimal power flow using simulated annealing," *International Journal of Electrical Power & Energy Systems*, vol. 25, no. 1, pp. 47-57, January 2003.

[11] H. Shin, M. Park, S. Yang, S, Moon, and S. Hong, "Locally optimal solution of robust ex-core detector response using constrained simulated annealing," *Nuclear Engineering and Design*, vol. 239, no. 1, pp. 51-57, January 2009.

[12] M. Duque-Anton, "Constructing efficient simulated annealing algorithms," *Discrete Applied Mathematics*, vol. 77, no. 2, pp. 139-159, August 1997.

[13] N. Manavidadeh, N. Hoseini, M. Rabani, and F. Jolai, "A simulated annealing algorithm for a mixed model assembly U-line balancing type-I problem considering human efficiency and Just-In-Time approach," *Computers& Industrial Engineering*, vol. 64, no. 2, pp. 669-685, February 2013.
[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," presented at the IEEE Latin-American Conference on Neural Networks, Perth, November 27-30, 1995.
[15] S. Tsafarakis, Y. Marinakis, and N. Matsasinis, "Particle swarm optimization for optimal product line design," *Journal of International Journal of Research in Marketing*, vol. 28, no. 1, pp. 13-22, March 2011.
[16] B. Liu, L. Wang, Y. Jin, F. Tang, and D. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons & Fractals*, vol. 25, no. 5, pp. 1261-1271, September 2005.
[17] F. Cheng, X. Sun, D. Wei, and Y. Tang, "Tradeoff strategy between exploration and exploitation for PSO," presented at the 2011 Seventh Annual Conference on Natural Computation, Shanghai, July 26-28, 2011.
[18] I. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317-325, March 2003.

Aaron Don M. Africa earned his B.S. Degree in Electronics and Communications Engineering (ECE) from the University of Santo Tomas. He completed his Masters Degree in ECE at the De La Salle University Manila. Presently, he is an Associate Professor of De La Salle University Manila and a Ph.D. graduate in ECE of the said school.