中正嶺學報 第三十八卷 第一期 民國 98.11.
JOURNAL OF C.C.I.T., VOL.38, NO.1, NOV., 2009

A Meta-heuristic Approach to Buffer Allocation in Production Line

Hsu-Tung Lee[1*], Shao-Kai Chen[1], and Shunder Chang[1]

[1] *Department of Business Administration, National Taipei University*

ABSTRACT

Choosing suitable buffer setups for production lines to augment throughputs is a pragmatic issue. Previous studies mainly centered on the buffer allocation problem with a single product in static machine layouts. When production lines have frequent product changes and setup adjustments, the buffer allocations have to be altered accordingly to meet the required production rate. Hence, it is more crucial to create a balance between effectiveness (by using the mathematical method) and efficiency (by using the heuristic method). Artificial intelligence (AI) is widely applied to accommodate many problems with appropriate solutions. In this study, an AI based method is proposed to investigate the buffer allocation of unbalanced-unreliable flow type production lines. Genetic Algorithm (GA) combined with simulation method is used in attempting to quickly figure out the best solutions. In turn, these optimal solutions are fed into an Artificial Neural Network (ANN) for predicting buffer layouts. Through the well-trained ANN, the preferable buffer allocation can be predicted promptly.

Keywords: genetic algorithm, simulation, artificial neural network, unbalance production line, unreliable production line

一般性流程生產線緩衝區配置之研究

李緒東[1*]　陳少開[1]　張舜德[1]

[1] 國立台北大學大學 企業管理學系

摘　要

當生產線面臨頻繁的產品調整或更改，如何配置適合的緩衝區大小以提昇產能是一個產業界亟需解決的問題。本研究使用人工智慧方法，來解決緩衝區動態配置的問題。首先使用基因演算法和模擬軟體搭配，快速找出一系列的最佳解，接著再將最佳解套入類神經網路做訓練，所產生的類神經網路模型可即時找出緩衝區最佳配置之情形。研究結果證實，經完整訓練的類神網路模型，可取代傳統方法快速且成功的找出接近最佳的緩衝區配置。

關鍵詞：基因演算法，模擬，類神經網路，不平衡生產線，故障生產線

文稿收件日期 97.12.18; 文稿修正後接受日期 98.10.13;*通訊作者
Manuscript received Dec 18, 2008; revised October 13;* Corresponding author

I. INTRODUCTION

A production line (transfer line) is a series of machines or workstations that are connected one-by-one and which perform sequential tasks. An ideal production line is able to produce maximum yield with great flexibility with frequent product change and unexpected order change. Two main concerns arise when production lines are constituted: the process time balance and the reliability of work centers. Basically, it is very rare to perfectly balance a production line in terms of process time and its variances. In addition, every production system is subject to stoppage, machine breakdown, unscheduled maintenance, job interruption, machine setup, and many other factors. To reduce the impact of stoppage and process time variation, manufacturers usually put buffers between workstation to maintain the integrity of production.

For smooth production, the intermachine buffers should maintain certain amounts of work-in-process (WIP) to neutralize the effect of machine stoppage. However, most of the shops contain limited space and it is not economical to keep too much WIP. Thus, properly arranging the intermachine buffer becomes a very important issue. Substantial research shows that carefully arranging buffers can achieve better production levels. Buzacott [1] uses Markov chain models to solve this problem. El-Rayah [2] studies the behavior of production lines with different interstage buffer assignments. Likewise, the study proves that assigning larger interstage buffers capacities to the middle workstation will be more efficient while unbalancing the production line in terms of buffer capacities. Choong and Gershwin [3] develop a decomposition method to reduce the state space in this problem and prove it to be accurate. Hillier and So [4] study the effects of machine breakdowns and interstage buffer capacity on performance of the production line. Kouikoglou and Phillis [5] present a simulation-based model for minimizing buffer size while meeting demand constraints for finite buffers and unreliable production line. Gershwin and Schor [6] propose a two-ply algorithm; first, to maximize the production rate, and second, to find out the minimum buffer allocation for the production rate. Papadopoulos and Vidalis [7] develop a heuristic method for unreliable and unbalanced production lines to find reasonable good initial buffer allocation. Sörensen and Janssens [8] form an unreliable machine, with finite buffer allocation problem as the non-linear programming problem and with cost and buffer usage minimization as the objectives. Macgregor and Cruz [9] develop closed-form expressions for production line with series, merge, and splitting topologies. Nahas *et al.* [10] use degraded ceiling search techniques to find the maximum throughput for an unbalanced-unreliable production line.

Research also applies soft computing-related methods to solve this problem. Bulgak *et al.* [11] and Wellman and Gemmill [12] use the GA method to study the buffer allocation in close-loop production line. Lutz *et al.* [13] combine the Tabu search and simulation methods to find out the maximum output for a given storage level. Spinellis and Papadopoulos [14] use simulated annealing and decomposition method to maximize the average throughput for buffer allocation problem in a reliable production line. Dolgui *et al.* [15] propose a GA and Markov-based model for finite buffer/unreliable production line buffer allocation problems. Shi and Men [16] present a hybrid method by combining Tabu search and nested partition for large production problem. Bulgak [17] extends previous studies, develops a GA-ANN-based metamodeling approach for split-and-merge asynchronous assembly systems (AAS) with fixed cycle time. Altiparmak *et al.* [18] further extend Bulgak's research into single close-loop AAS with three different machine failure occurrences: same failure rate, zero and non-zero failure rate, and low- and high failure rate, respectively. Lee and Wang [19] apply the GA-based method in a just-in-time environment with unreliable machines and investigate the minimum kanban required.

Inasmuch as a production line needs to frequently adapt to the changing environment, it is essential to decide a suitable buffer layout at a desirable production rate in a very short period of time. In this research, the GA-ANN-based method developed by Bulgak *et al* [11] has been modified to extend Lee and Wang's [19] studies. The studied problem is a general finite buffer production line with total buffer limitation, in which the machine breakdown and production rate varies according to products and a suitable buffer allocation has to be decided quickly.

II. PROBLEM STATEMENT

The system has N unreliable workstations and N-1 interstage buffers. This is shown as Figure 1. The interstage buffer is utilized to neutralize the possible stoppage and the capacities (quantity of each buffer) need to be adjusted according to product assignment.

Some of the characters for this production line are listed as follows:

1. All items enter at the first workstation and leave at the last workstation.
2. All machines are subject to breakdown.
3. Each machine has product-dependent production and breakdown rates.
4. The total buffer space is limited.
5. There is an infinite source for the production system, and an infinite storage capacity at the end. In other words, the first machine never starves, and the last machine can never be blocked.
6. Blockage (starvation) exists when the machine in the upstream (downstream) is ready but there is no kanban to authorize production activity. This situation may occur when the downstream (upstream) machine breaks down and the buffers are full (empty). Breakdown cannot occur at a starved or blocked workstation.
7. The process time, failure rates, and recovery rates of workstations and their endurance follow statistical distributions.

Intuitively, the throughput will reach the maximum if the interstage buffers are infinite. However, it is not economical. The objective of this research is to find out the minimum buffer sizes associated with workstation that maximize throughput with respect to total buffer space constraints for each product. For each product, the problem can be formulated as a two-phase optimization problem as follows:

Phase 1
Maximize *THR*
Subject to:

$$\sum_{i=1}^{n-1} B_i \leq K$$

$$B_i \leq k_i$$

(production rate limitation for machine i)
(breakdown rate for machine i)

Where:
THR is the throughput of each product
B_i is the buffer size between machine *i* and *i+1*
K is the total buffer space
k_i is the maximum buffer allowed for B_i

Phase 2

Minimize $\sum_{i=1}^{n-1} B_i$ (*total buffer used*)
Subject to:
(*System throughput unchanged from phase 1*)

It is difficult to develop a heuristic or analytical solution for this NP-hard combinatorial optimization problem (Garey and Johnson [20]). In addition, flexible adaption to product changes and quick adjustment of the buffer sizes to achieve maximum throughput make the problem even more difficult.

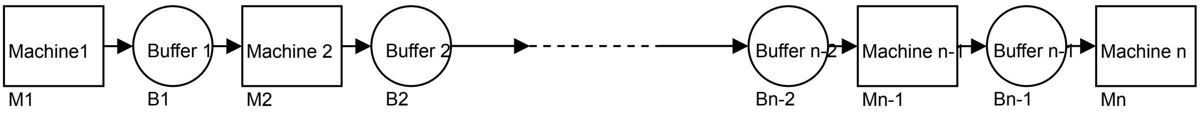


Fig. 1. The topology of the production system.

III. METHODOLOGY

Inspired by Bulgak's research [11, 17], a soft computing-based two-stage algorithm is proposed to achieve the goal of this research. The two-stage algorithm is described as follows:

Table 1. Research methodology

Stage	Step
1. Searching for buffer allocation for each product	1. Randomly generate populations for each production line. 2. Build up a simulation model for each buffer layout; simulate the production rate. 3. Apply GA operations for new buffer layout. . 4. Repeat steps 2 to 3 until optimal or near optimal buffer layout have been found. 5. Repeat steps 1 to 4 for all products.
2. Build an ANN to predict buffer allocation	6. Build up an ANN by using the results from step 5. 7. Train the ANN. 8. Validate the ANN.

The flowchart of this algorithm is depicted in Figure 2. The feature of each stage is described in Sections 3-1 and 3-2; a numerical example with detail is illustrated in Sections 4-1 and 4-2.

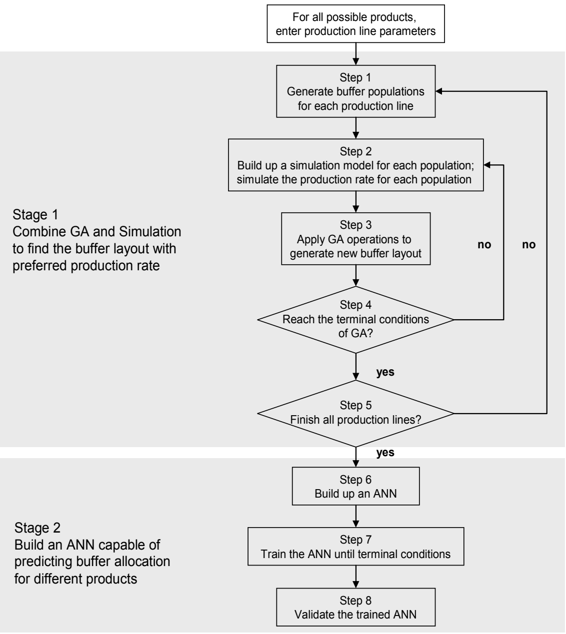


Fig. 2. Flowchart of the Buffer Allocation Searching algorithm.

3.1 Stage 1: Searching for buffer allocation for each product

The state space of the aforementioned phase 1 optimization problem has NS number of states; and

$$NS = \frac{(K+N-2)!}{(K-N+1)!\,(2N-1)!}$$

when non-zero individual buffer is assumed. This research proposes a non-mathematical approach other than the conventional decomposition method by combining the advantages of GA and simulation method. The simulation models serves as the fitness function of GA and an improved algorithm is used to expedite the search processes of GA while keeping the buffer at a minimum. The details can be described as follows:

Step 1: For each product, where the production rate and breakdown rate are assigned, randomly generate a group of buffer layouts where the total summation of buffer should not exceed the total buffer limitation.

Step 2: Build up a simulation model to simulate the throughput for each buffer layout.

Step 3: Use the GA operators to generate new offspring (buffer layouts) with better throughput.

Step 3.1: Use binary code to represent the size of buffer.

Step 3.2: Use elitist strategy mentioned by Bean [21] and roulette wheel selection as the reproduction operator.

Step 3.3: Use two-point crossover to generate offspring.

Step 3.4: Apply buffer adjustment while the result from crossover exceeds total buffer limitation.

Step 3.5: Use a suitable mutation rate as Bean [21] recommends.

Step 4: Repeat steps 2 and 3 until terminating condition is reached.

Step 5: Repeat steps1 through 4 for all possible products.

3.2 Stage 2: Build an ANN to predict buffer allocation

ANN simulates human information passing behavior with artificial neurons interconnecting with weights. By adjusting the associated weights, ANN can transform a set of inputs into a set of desired outputs. The method is widely adopted into many applications and provides satisfactory results. In this research, an ANN is trained and used to predict the desired buffer layout. The steps are as followed.

Step 6: Build up an ANN where the inputs are: production rate, and breakdown rate; the outputs are buffer allocation.

Step 7: Train the ANN with partial results from Step 5.

Step 8: Validate the ANN with the remaining results from Step 5.

IV. NUMERICAL EXAMPLE

4.1 Numerical example 1

In this example, the result from GA-simulation based method is compared to the results from benchmark researches. The benchmark researches chosen here are: Ho *et al.* [22] which was among the earliest to introduce perturbation analysis in transfer line, and Gershwin and Schor [6] which used primal and dual method to consider maximizing production rate and minimizing buffer size simultaneously. A five-machine, four-buffer, unreliable production line is considered. The parameters of the production line are listed in Table 2.

The GA-simulation based method described in section 3-1 is used. The details of each step are listed as followed.

Step 1: Randomly generate 30 sets of buffer layout as the initial population; the individual buffer limitation is set to be 15.

Step 2: Use simulation software (Flexsim 4.3) to model 30 production lines according to the buffer layout generated in Step1; simulate each production line for 5000 minutes which require less than 2 minutes in computer time; obtain the throughput for all production lines.

Step 3: The offspring of buffer layout are generated according GA operations, and the throughputs obtained in Step 2 are the fitness of production lines.

Step 3.1: Coding: Use binary coding for each buffer layout. Since the individual buffer limitation is 15, by using binary coding schema, the length of each buffer is 4, and the chromosome length of the four-buffer system is 16. Figure 3 illustrated the coding schema for a {7, 10, 10, 4} buffer allocation.

Step 3.2: Selection: Use roulette wheel and elitist strategy for mating selection. The elitist percent is set to be 20%, in other word, the production line with top 20% of fitness will enter the mating pool automatically

Table 2. Parameters of the five-machine benchmark problem

	Machine parameter				
	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
MTBF	20	167	22	22	26
MTTR	11	19	12	7	7

	Buffer 1	**Buffer 2**	**Buffer 3**	**Buffer 4**
Buffer size	7	10	10	4
Binary code	0111	1010	1010	0100

Fig. 3. Illustration of a 6-8-14-10-2 buffer coding.

Parent 1	9	4	5	4
Parent 2	5	6	7	8
Randomly select crossover point, for example 2 and 3				
Offspring 1	9	6	7	8
Offspring 2	5	4	5	4

Fig. 4. Illustration of two-point crossover operation.

Step 3.3: Crossover: Use two-point crossover method with 0.6 crossover rate. The two-point crossover method is illustrated in Figure 4.

Step 3.4: Buffer adjustment: Since the examined problem has no total buffer limitation, there is no need for buffer adjustment.

Step 3.5: Mutation: Use 0.033 mutation rate.

Step 4: Repeat Steps 2 and 3 until terminating condition is reached. The termination conditions are as followed.

- Minimum generation= 15, and maximum generation= 150
- Best fitness sought unchanged for 5 generations and the difference between the best fitness and the average of fitness is less than 0.5%
- Best fitness sought unchanged for 10 cycles

Since the selected chromosome with best fitness could be draw from elitist group for generations, the later two conditions are built to allow enough mixture for the crossover operation and prevent early stop. The flowchart of terminating conditions is depicted in Figure 5.

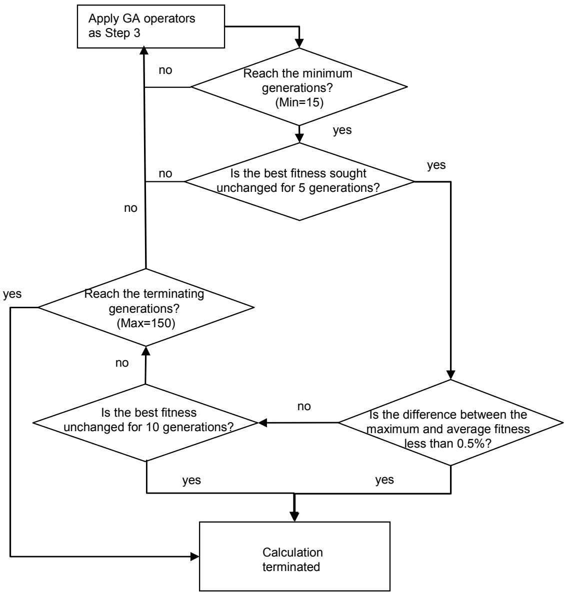


Fig. 5. The terminating conditions of GA..

Step 5: Repeat Steps 1 to 4 for all possible product. There is only one product in the benchmark research, thus the calculation terminated here.

The optimal buffer allocations suggested by Ho *et al.'s* and Gershwin and Schor's algorithms are {5, 11, 8, 7} and {7, 10, 10, 4}, respectively. The output for buffer layout {7, 10, 10, 4} is 3119 units. In contrast, the buffer layout {5, 11, 8, 7} is 3092 units. Applying the GA-simulation algorithm from Step 1 through 4, the optimal/near optimal solution is obtained in generation 15 and the four buffers are {7, 10, 10, 4}. The output of proposed algorithm is coherent with Gershwin and Schor's result and it out performs the result from Ho *et al.'s*.

4.2 Numerical example 2

A six-machine five-buffer production line appeared in Bulgak's research is chosen for the buffer allocation ANN building. The other characteristics of this production line are listed as follows:

- Product type: 100 products of different production rate combination are considered.
- Production rate: cycle times of machines for different product are randomly generated between two to four minutes.
- Machine breakdown rate: 2% and 4% of breakdown rate are randomly assigned to each machine.
- Machine failure clear time: is set at 50 minutes.
- Buffer limitations: the total buffer limitation is set at 40, and the individual buffer limitation is set at 15.

The details for each step are listed as followed.

Stage 1: Searching for buffer allocation for each product

Step 1: Generate 30 populations under the total buffer limitation for each product.

Step 2: Build model for each population; simulate for 5000 minutes; obtain the throughput for all production lines.

Step 3: Generate offspring.

Step 3.1: Coding: Use binary coding for the five-buffer system; the length of chromosome is 20.

Step 3.2: Selection: Use roulette wheel and elitist strategy for mating selection. The elitist percent is set to be 20%.

Step 3.3: Crossover: Use two-point crossover method with 0.6 crossover rate.

Step 3.4: Buffer adjustment: The crossover operation could generate buffer layouts exceed the total buffer limitation, thus buffer adjustment must be performed. For example, a {15, 12, 4, 8, 10} buffer layout after crossover has the total buffer of 48, which exceeds the buffer limitation of 40. The modified buffer according to weight is {12, 10, 3, 6, 8}.

Step 3.5: Mutation: Use 0.033 mutation rate.

Step 4: Repeat Steps 2 and 3 until terminating condition is reached. The termination conditions are the same as the previous example. Table 3 lists one of the parameter set of the production lines, and Figure 6 demonstrates its solution procedures of GA. Notice that the calculation procedures reach stopping condition at the 18th generation because of the best outputs that remained unchanged for five generations. In addition, the average output of generation 18 is 2461.2 and the difference is within the predetermined range (=0.5%) to the best output. Table 4 lists the best results of each generation.

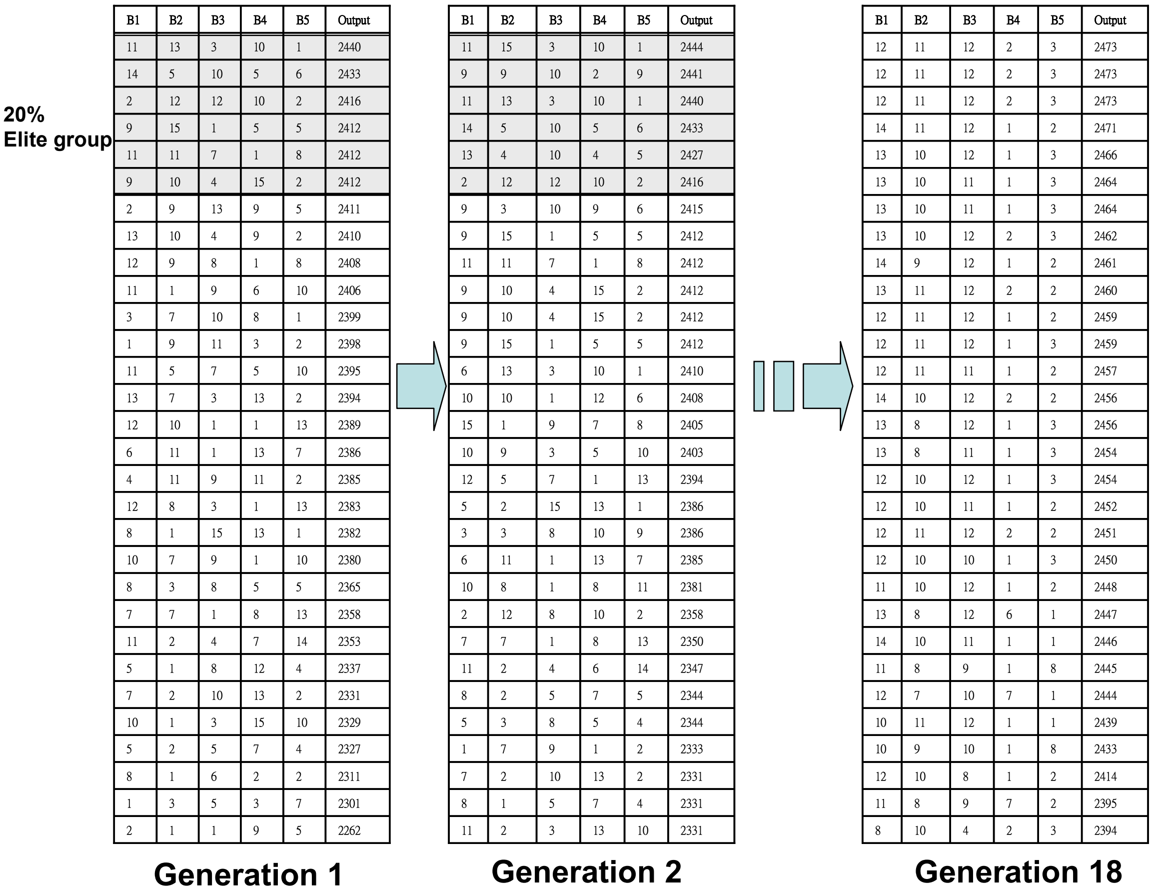
Step 5: Repeat Steps 1 to 4 for all 100 products.

Stage 2: Fast predict buffer allocation and result validation

Step 6: A three-layer back propagation network is built for this example, and the Neural Network Toolbox 4.0 in MATLAB 7.0 is used to assist the model building. The input layer consists of 12 neurons and they are the production rate and breakdown rate for each machine. The hidden layer contains 10 neurons. The output layer is the buffer sizes (five neurons). A momentum term was considered in this study to avoid trapping in the local minimum. The total sum of squared error (TSS) is set at 0.05 and the maximum iteration is set at 100.

Table 3. Sample Machine Parameters

	Machine parameter					
	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5	Machine 6
Cycle time	3	3.9	3.3	2.9	3.4	3
Breakdown rate	4%	4%	4%	2%	4%	2%



Generation 1

B1	B2	B3	B4	B5	Output
11	13	3	10	1	2440
14	5	10	5	6	2433
2	12	12	10	2	2416
9	15	1	5	5	2412
11	11	7	1	8	2412
9	10	4	15	2	2412
2	9	13	9	5	2411
13	10	4	9	2	2410
12	9	8	1	8	2408
11	1	9	6	10	2406
3	7	10	8	1	2399
1	9	11	3	2	2398
11	5	7	5	10	2395
13	7	3	13	2	2394
12	10	1	1	13	2389
6	11	1	13	7	2386
4	11	9	11	2	2385
12	8	3	1	13	2383
8	1	15	13	1	2382
10	7	9	1	10	2380
8	3	8	5	5	2365
7	7	1	8	13	2358
11	2	4	7	14	2353
5	1	8	12	4	2337
7	2	10	13	2	2331
10	1	3	15	10	2329
5	2	5	7	4	2327
8	1	6	2	2	2311
1	3	5	3	7	2301
2	1	1	9	5	2262

Generation 2

B1	B2	B3	B4	B5	Output
11	15	3	10	1	2444
9	9	10	2	9	2441
11	13	3	10	1	2440
14	5	10	5	6	2433
13	4	10	4	5	2427
2	12	12	10	2	2416
9	3	10	9	6	2415
9	15	1	5	5	2412
11	11	7	1	8	2412
9	10	4	15	2	2412
9	10	4	15	2	2412
9	15	1	5	5	2412
6	13	3	10	1	2410
10	10	1	12	6	2408
15	1	9	7	8	2405
10	9	3	5	10	2403
12	5	7	1	13	2394
5	2	15	13	1	2386
3	3	8	10	9	2386
6	11	1	13	7	2385
10	8	1	8	11	2381
2	12	8	10	2	2358
7	7	1	8	13	2350
11	2	4	6	14	2347
8	2	5	7	5	2344
5	3	8	5	4	2344
1	7	9	1	2	2333
7	2	10	13	2	2331
8	1	5	7	4	2331
11	2	3	13	10	2331

Generation 18

B1	B2	B3	B4	B5	Output
12	11	12	2	3	2473
12	11	12	2	3	2473
12	11	12	2	3	2473
14	11	12	1	2	2471
13	10	12	1	3	2466
13	10	11	1	3	2464
13	10	11	1	3	2464
13	10	12	2	3	2462
14	9	12	1	2	2461
13	11	12	2	2	2460
12	11	12	1	2	2459
12	11	12	1	3	2459
12	11	11	1	2	2457
14	10	12	2	2	2456
13	8	12	1	3	2456
13	8	11	1	3	2454
12	10	12	1	3	2454
12	10	11	1	2	2452
12	11	12	2	2	2451
12	10	10	1	3	2450
11	10	12	1	2	2448
13	8	12	6	1	2447
14	10	11	1	1	2446
11	8	9	1	8	2445
12	7	10	7	1	2444
10	11	12	1	1	2439
10	9	10	1	8	2433
12	10	8	1	2	2414
11	8	9	7	2	2395
8	10	4	2	3	2394

Fig. 6. The generational results of GA procedures.

Step 7: Randomly select 80 sets of results obtained from previous steps as the training group.

Step 8: The remaining sets are served as the testing group. The buffer allocations predicted by the trained ANN and their results from GA-simulation method are listed in Table 5. The throughputs for each buffer layout are quite close. The worst prediction is in group 17, while the difference is reasonably small at 2.3%. Figure 7 shows a best linear-fit regression between the outputs of the two methods. An F-test for the lack of fit verified the adequacy of the regression model performed and showed no significant difference between the results.

V. CONCLUSION

Nowadays, it is very important to promptly adapt to production changes while maintaining adequate outputs. Traditional methods for buffer allocation are either time consuming or problem-oriented. The GA-simulation-oriented method can produce effective result for most buffer allocation problems. Traditionally, it needs a lengthy period of time to complete a calculation routine for a six-machine buffer allocation problem on a Pentium 4 personal computer and the problem becomes more complicated when product increases. Through the proposed method, a well-trained ANN can accurately predict the buffer allocation. Shop engineers can simply enter the production rate and maintenance (breakdown) rate of a product and the ANN can provide satisfactory buffer layout in a very short period of time.

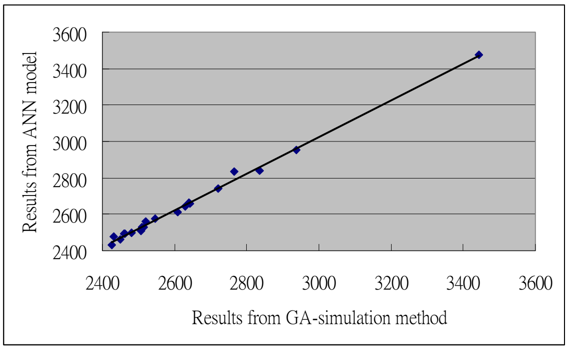


Fig. 7. Regression analysis of results from two methods.

Table 4. The best results of each GA generation

Generation	Buffer allocation					Max. Throughput
	B1	B2	B3	B4	B5	
1	11	13	3	10	1	2440
2	11	15	3	10	1	2444
3	11	15	3	10	1	2444
4	9	11	10	2	5	2451
5	9	11	10	2	5	2451
6	12	8	12	2	3	2458
7	12	8	12	2	3	2458
8	15	11	10	1	2	2467
9	15	11	10	1	2	2467
10	12	10	12	2	3	2468
11	12	10	12	2	3	2468
12	15	11	11	1	2	2469
13	12	10	13	2	3	2470
14	12	11	12	2	3	2473
15	12	11	12	2	3	2473
16	12	11	12	2	3	2473
17	12	11	12	2	3	2473
18	12	11	12	2	3	2473

Table 5. The outputs and buffer allocation comparison

Group	Buffer allocation predicted by ANN						Buffer allocation obtained from stage 1					
	B1	B2	B3	B4	B5	Output	B1	B2	B3	B4	B5	Output
1	3	12	11	6	8	2609	4	10	8	9	9	2612
2	5	6	13	10	6	2643	2	11	10	11	5	2658
3	6	12	8	2	12	2480	5	12	11	4	7	2499
4	4	7	11	12	6	2836	1	8	10	14	7	2839
5	6	13	12	6	3	2507	6	14	12	3	5	2507
6	3	8	13	12	4	2461	3	10	9	14	3	2495
7	8	7	11	7	7	2937	9	4	11	9	7	2951
8	10	8	8	8	5	2547	5	8	13	10	4	2575
9	9	10	10	8	3	2629	11	13	7	8	1	2641
10	7	11	8	9	5	2515	6	8	13	10	3	2531
11	4	9	9	6	11	2639	1	3	14	9	11	2665
12	6	13	10	10	1	2510	3	10	15	9	1	2527
13	8	9	10	8	4	2519	5	12	5	9	8	2558
14	2	13	11	6	8	2721	1	13	13	11	1	2740
15	7	13	12	5	2	2450	5	14	15	4	2	2462
16	11	10	9	7	3	3443	3	13	5	13	5	3477
17	3	11	12	4	9	2766	2	9	7	14	6	2832
18	7	4	11	12	6	2463	10	1	13	12	4	2492
19	7	9	10	9	3	2427	7	10	9	9	1	2431
20	10	7	8	8	7	2432	10	5	7	12	5	2478

VI. REFERENCES

[1] Buzacott, J. A., “Automatic Transfer Lines with Buffer Stocks,” International Journal of Production Research, Vol. 5, No. 3, pp. 182-200, 1967.

[2] El-Rayah, T. E., “The effect of inequality of interstage capacities and operation time variability on the efficiency of production line systems,” International Journal of Production Research, Vol. 17, pp. 77-89, 1979.

[3] Choong, Y. F., and Gershwin, S. B., “A Decomposition Method for the Approximate Evaluation of Capacitate Transfer Lines with Unreliable Machines and Random Processing Times,” IIE Transactions, Vol. 19, No. 2, pp. 150-159, 1987.

[4] Hillier, F. S. and So, K. C., “The effect of machine breakdowns and interstage storage on the performance of production line systems,” International Journal of Production Research, Vol. 29, No. 10, pp. 2043-2055, 1991.

[5] Kouikoglou, V., and Phillis, Y., “Continuous flow model for production networks with finite buffers, unreliable machines and multiple products,” International Journal of Production Research, Vol. 35, No. 2, pp. 381–397, 1997.

[6] Gershwin, S. B. and Schor, J. E., “Efficient algorithms for buffer space allocation,” Annals of Operations Research, Vol. 93, pp. 117-144, 2000.

[7] Papadopoulos, H. T., and Vidalis, M. I., “A heuristic algorithm for the buffer allocation in unreliable unbalanced production lines,” Computers & Industrial Engineering, Vol. 41, pp. 261-277, 2001.

[8] Sörensen, K., and Janssens, G. K., “Buffer allocation and required availability in a transfer line with unreliable machines,” International Journal of Production Economics, Vol. 74, pp. 163-173, 2001.

[9] MacGregor S. J., and Cruz, F. R. B., “The buffer allocation problem for general finite buffer queueing networks,” IIE Transactions, Vol. 37, pp. 343-365, 2005.

[10] Nahas, N., Ait-Kadi, D. and Nourelfath, M., “A new approach for buffer allocation in unreliable production lines,” International Journal of Production Economics, Vol. 103, pp. 873-881, 2006.

[11] Bulgak, A. A., Diwan, P. D. and Inozu, B., “Buffer size optimization in asynchronous assembly systems using genetic algorithms,” Computers & Industrial Engineering, Vol. 28, pp. 309-322, 1995.

[12] Wellman, M. A. and Gemmill, D. D., “A genetic algorithm approach to optimization of asynchronous automatic assembly systems,” The international journal of Flexible Manufacturing systems, Vol. 7, pp. 27-46, 1995.

[13] Lutz, C. M., Davis, K. R. and Sun, M., “Determining buffer location and size in production lines using tabu search,” European Journal of Operational Research, Vol. 106, pp. 301-316, 1998.

[14] Spinellis, D. D., and Papadopoulos, C. T., “A simulated annealing approach for buffer allocation in reliable production line,” Annals of Operations Research, Vol. 93, pp. 373-384, 2000.

[15] Dolgui, A., Eremeev, A., Kolokolov, A., and Sigaev, V., “A Genetic Algorithm for the Allocation of Buffer Storage Capacities in a Production Line with Unreliable Machines,” Journal of Mathematical Modelling and Algorithms, Vol. 1, pp. 89–104, 2002.

[16] Shi, L. and Men, S., “Optimal buffer allocation in production lines,” IIE Transactions, Vol. 35, pp. 1-10, 2003.

[17] Bulgak, A. A., “Analysis and design of split and merge unpaced assembly systems by metamodelling and stochastic search,” International Journal of Production Research, Vol. 44, pp. 4067-4080, 2006.

[18] Altiparmak, F., Dengiz, B. and Bulgak, A. A., “Buffer allocation and performance modeling in asynchronous assembly system operations: An artificial neural network metamodeling approach,” Applied Soft Computing, Vol. 7, pp. 946-956, 2007.

[19] Lee, H., and Wang, M., “On the search of workstations arrangement in pull production systems,” Computers & Industrial Engineering, Vol. 54, No. 3, pp. 613-623, 2008.

[20] Garey, M., and Johnson, D., Computers and Intractability, San Francisco, CA., W. H. Freeman, 1979.

[21] Bean, J. C., “Genetic algorithms and random keys for sequencing and optimization,” ORSA Journal on Computing, Vol. 6 No 2, pp. 154-160, 1994.

[22] Ho, Y. C., Eyler, M. A. and Chien, T. T., “A new approach to determine parameter sensitivities of transfer lines,” Management science, Vol. 29, No. 6, pp. 700-714, 1983.