Expert Systems with Applications 38 (2011) 5912–5918

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An integrated MOGA approach to determine the Pareto-optimal kanban number and size for a JIT system

Tung-Hsu (Tony) Hou *, Wei-Chung Hu

Department of Industrial Engineering and Management, National Yunlin University of Science and Technology, 123 University Rd., Sec. 3, Douliu, Yunlin 640, Taiwan, ROC

ARTICLE INFO

Keywords:
JIT
Lean production
Kanban
MOGA

ABSTRACT

In a just-in-time (JIT) system, kanban number and size represent the inventory level of work-in-process (WIP) or purchasing parts. It is an important issue to determine the feasible kanban number and size. In this research, an integrated multiple-objective genetic algorithm (MOGA) based system is developed to determine the Pareto-optimal kanban number and size, and is applied in a JIT-oriented manufacturing company to demonstrate its feasibility. In the integrated system, a simulation model is built to simulate the multi-stage JIT production system of the company. Then an experimental design of different kanban numbers and sizes for different production stages is applied to test the production performances. Based on the simulation results, regression models are built to represent the relationships between the kanban numbers of different production stages and the production performance. These regression models are then used in genetic algorithms to generate the performance for chromosomes. Finally, the proposed multi-objective genetic algorithm (MOGA) based system uses the generalized Parato-based scale independent fitness function (GPSIFF) as the fitness function to evaluate the multiple objectives for chromosomes and used to find the Pareto-optimal kanban number and size for multiple objectives, i.e., maximizing mean throughput rate and minimizing mean total WIP inventory. A comparison in the performance of the proposed system with that of the current kanban number is conducted to demonstrate the feasibility of the proposed system.

1. Introduction

The just-in-time (JIT) production system has been applied widely in many manufacturing enterprise around the world, mainly due to the success of Toyota Motor Company. The use of JIT system can result in minimizing the inventory level and manufacturing lead time, and simultaneously achieving high quality level and customer satisfaction. The underlying principle of JIT philosophy is to produce the right quantity of product at the right time with the right quality level. Kanban, which means a card in Japanese, is a tool used to achieve JIT production.

In a JIT system, production is triggered by a kanban signal, which usually comes from the customer order. The signal then flows backward from the final assembly station to the upstream production centers, and then to the suppliers. Each work-in-progress (WIP) container is attached with a kanban specifying the details of that WIP such as the part name, part number, downstream process, upstream process, container size, the maximum kanban number, etc. The container size is equivalent to the kanban size and the kanban number represents the WIP container number. A large kanban size and kanban number represent a high WIP level.

In manufacturing practice, the kanban size is usually assumed to be fixed and the kanban number is computed through empirical equations. Monden (1993) indicated that kanban number under constant quantity withdrawal system is the dividing of the multiplication of daily demand, lead time and safety factor by the container size. These empirical equations may result in higher inventory level. In addition, the kanban size needed to be carefully set to minimize the WIP and achieve customer satisfaction.

Chan (2001) indicated that kanban size did have effects on JIT manufacturing system performance. For multiple product production, as the kanban size increased, the fill rate increased with a decrease in manufacture lead time. Yavuz and Stair (1995) concluded that reducing the kanban size could reduce the inventory levels and the makespan. On the other hand, increasing the kanban size will increase WIP, but improve the fill rate. Therefore, reducing the kanban size to achieve lower inventory level, and simultaneously retaining the full customer satisfaction (i.e., fill rate) may not be easily implemented in real situations. The kanban size is a vital problem in JIT system which is worthy of being investigated.

Since using the empirical equation to compute the kanban number could result in too many WIPs, many researchers have

* Corresponding author. Tel.: +886 5 5342601; fax: +886 5 5312073.
E-mail address: houth@yuntech.edu.tw (Tung-Hsu (Tony) Hou).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2010.11.032

focused on findings the optimal kanban number in production system. Some researchers tried to find the optimal kanban number based on a cost model. Nori and Sarker (1998) constructed a mathematical cost model for two adjacent stations, and minimized the total cost to find the optimal kanban number. They considered the inventory cost and shortage cost, and applied the incremental search procedure to find the optimal solution. Wang and Sarker (2006) constructed a nonlinear programming model for a multi-stage supply chain system controlled by kanban, and minimized the total cost. Branch-and-bound algorithm was applied to find the optimal kanban number.

Although these mathematical cost models could find the optimal kanban number, it usually combined multi-cost factors using different weights into a total cost. The weight of each cost factor was given. Once the weight of a cost factor is changed, the model will generate different optimal number. In addition, the decision maker of the plant was only given one optimal solution, instead of a set of solutions which showed the trade-off among the cost factors, such as the trade-off between the cost of set-up or material handlings and the benefits of lower WIP levels. To find the optimal kanban number that shows the trade-off in different factors is another vital problem in JIT system. This kind of problem is a multi-objective optimization problem, and multi-objective genetic algorithm (MOGA) can solve this kind of problem. Mansouri (2005) successfully applied MOGA to determine the mixed production sequences in a JIT assembly line. However, none a MOGA application was found in determining the kanban number and kanban size in a JIT system.

Some researchers used simulation to find the effects of kanban number on production system performance. Savsar and Al-Jawini (1995) built a simulation model for a JIT production system and found that different kanban rules resulted in different production performance in WIP and throughput rate. Duri, Frein, and Di Mascolo (1995) constructed a simulation model for a mixed-model production system. They found as kanban number increased, throughput rate and WIP increased, but waiting time and proportion of backorder decreased. These simulation models did reveal the effects of kanban number on production performance, but they were not used to find the optimal kanban number. Recently, Köchel and Nieländer (2002) combined simulation and genetic algorithm to determine the optimal kanban number; however, they combined multiple cost factors into one objective function and generated only one optimal solution. They did not generate a set of solutions that showed the trade-off on the cost factors.

In addition, there is a trade-off between the kanban size and kanban number. When the kanban size is larger, the kanban number becomes smaller. It is necessary to determine the kanban size and kanban number simultaneously.

Recently, some researchers investigated the optimal flexible kanban number issue. Gupta and Al-Turki (1997) presented the flexible kanban system which dynamically determined the kanban number for the production system with uncertainties of processing time and variable demand. Lee (2007) presented a two-stage tabu search algorithm to determine the flexible kanban number by simultaneously considering kanban controlling and scheduling. He also showed that the simultaneous kanban controlling and scheduling could result in 30% cost reduction over scheduling without a kanban controlling. Guneri, Kuzu, and Taskin Gumus (2008) used a case study to verify the effectiveness of a flexible kanban system, and used an artificial neural network based simulation to confirm the advantage of the flexible kanban system over the traditional kanban system. Sivakumar and Shahabudeen (2008) employed a genetic algorithm and a simulated annealing algorithm to set the flexible kanban number for multi-stage flexible kanban system. They found that the flexible kanban system had better performance than the traditional kanban system, and the simulated algorithm generated better results that the genetic algorithm. However, they did not apply MOGA or multi-objective simulated algorithm to determine the kanban number.

Although the researchers put their research interests on the flexible kanban system recently, the issue of determining the kanban number and kanban size simultaneously for the traditional kanban system is still remained to be investigated. Therefore, the objective of this research is to propose an integrated multi-objective genetic algorithm based system to determine the optimal kanban size and kanban number for a multi-stage JIT system. The proposed system can find the feasible kanban size and kanban number for the decision maker and show the trade-off among the objectives for kanban size and kanban number combinations.

2. Multi-objective evolutionary algorithms

Problems that have two or more conflicting objectives to be simultaneously optimized are common in real-world applications. Such problems are called "multi-objective" or "multi-criteria" optimization problems. When dealing with the multi-objective optimization problem, the notion of "optimality" needs to be extended. The most common notion is called Edgeworth-Pareto optimality, or simply Pareto-optimal solutions, and refers to finding good trade-offs among all the objectives. This definition leads us to find a set of solutions that is called the Pareto-optimal set, whose corresponding elements are called the non-dominated solutions or non-inferior solutions. The Pareto-optimal set under the objective functions is called the Pareto-front.

The multi-objective evolutionary algorithms (MOEAs) are widely used in solving multi-objective optimization problems. The growing popularity of evolutionary algorithms is mainly due to their flexibility to deal with numerical and combinatorial multi-objective optimization problems and their ease of use. Also, due to their population-based nature, evolutionary algorithms can be modified to generate several Pareto-optimal (non-dominated) solutions in a single run.

Most MOEA algorithms use the concept of "domination" to find the Pareto-optimal solutions. Deb (1999) defined the dominate solution by the following rule:

A solution $X_{(1)}$ is said to dominate the other solution $X_{(2)}$ (In other words, $X_{(1)}$ is a non-dominated solution), if both the following conditions are satisfied:

Condition 1: The solution $X_{(1)}$ is no worse than $X_{(2)}$ in all objectives.

Condition 2: The solution $X_{(1)}$ is strictly better than $X_{(2)}$ in at lease one of the objectives.

If one of the above conditions is not satisfied, the solution $X_{(1)}$ does not dominate the solution $X_{(2)}$. For example, let us consider a two-objective (Max–Min) optimization problem with five different solutions in the objective space, as illustrated in Fig. 1. In this example, objective 1 is to be maximized while objective 2 is to be minimized. Since both objectives are equally important to the decision maker, it is usually difficult to find a solution that is the best with respect to these two objectives. The domination concept is used to determine the better solution between any two solutions in terms of both objectives. For instance, when solutions A and C are compared, solution C is strictly better than solution A in both objectives 1 and 2. In this way, both of the domination conditions are satisfied. Therefore, solution C dominates solution A. Let us take another example by comparing solution A and solution E. Here, solution E is strictly better than solution A in objective 1 and solution E is no worse than solution A in objective 2. Therefore, both of the domination conditions are also satisfied, and solution E

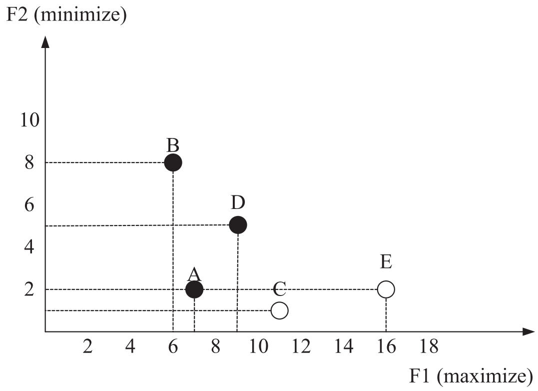

Fig. 1. An example of Pareto solution for a two-objective optimization problem with five solutions.

is said to dominate solution A. Using the domination concept to compare the solutions in multiple objectives, most MOEAs can find all of the non-dominated solutions that are also called the Pareto-optimal solution.

One numerical way to determine the Pareto-optimal solution is called the generalized Pareto-based scale-independent fitness function (GPSIFF). The basic idea of GPSIFF is that it evaluates the domination of each solution using a score function. A tournament-like score function is taken to find the non-dominated solutions. The score value of a solution $\boldsymbol{X}$ is calculated according to the following score function (Ho, Shu, & Chen, 2004):

$$\boldsymbol{score}(\boldsymbol{X}) = \boldsymbol{p} - \boldsymbol{q} + \boldsymbol{c}, \quad (1)$$

where, $\boldsymbol{p}$ is the number of solutions dominated by $\boldsymbol{X}$, and $\boldsymbol{q}$ is the number of solutions which dominate $\boldsymbol{X}$. The scaling constant $\boldsymbol{c}$ is used to get a positive fitness value.

By calculating the score values for all solutions, the solutions that have the highest score values can be found. Let us take the five solutions in the Fig. 1 as an example again and set constant $\boldsymbol{c}$ to 5. For the solution A, it is found that one solution (B) is dominated by A (i.e., $p = 1$) and two solutions (C and D) dominate A (i.e., $q = 2$). Therefore, the score value of A is $1 - 2 + 5 = 4$. By this way, we can calculate the score value for all solutions and find the Pareto-optimal solutions that have the highest score value. For the case in Fig. 1, score $(B) = 0$, score $(C) = 8$, score $(D) = 4$, and score $(E) = 8$. Therefore, solution C and E are the two Pareto-optimal solutions. In this study, the GPSIFF concept is applied to determine the Pareto-optimal solutions for the kanban size and kanban number in a JIT system.

3. The proposed integrated system

In the proposed system, a simulation model is built to simulate the multi-stage JIT production system of the company. Then an experimental design of different kanban numbers and sizes for different production stages is applied to test the production performances. Based on the simulation results, regression models are built to represent the relationships between the kanban numbers and kanban size of different production stages and the production performance. These regression models are then used in genetic algorithms to generate the performance for chromosomes. Finally, the proposed MOGA based system uses the GPSIFF as the fitness function to evaluate the multiple objectives for chromosomes and used to find the Pareto-optimal kanban number and size for multiple objectives.

3.1. Simulation model of a JIT system

In this research, Simprocess Release 4.3 is used to build the simulation model for a sun roof production line in the case company that applies JIT system to produce automobile parts. This company uses empirical equation to determine the kanban number for each station. Value stream mapping (VSM) of the sun roof production line is shown in Fig. 2.

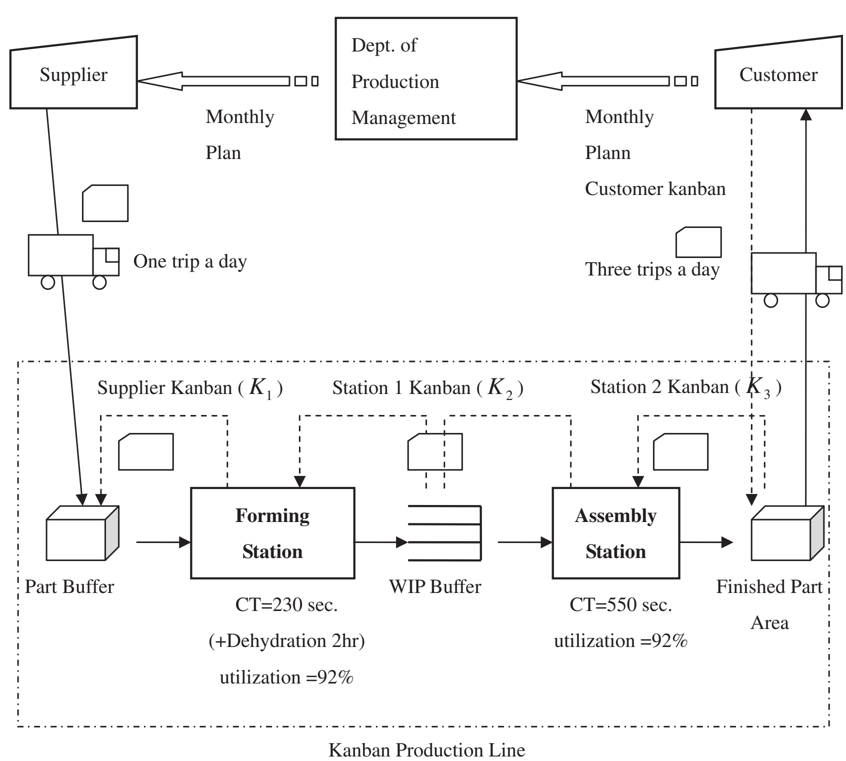

Fig. 2. Value stream mapping of the sun roof JIT production line.

The case company uses the two-card kanban system for the finished parts, and uses the one-card kanban system for the WIP and supplier parts. The customer issues a monthly production plan and daily customer kanbans to the case company that then delivers finished parts to his customer three times a day according to the daily customer kanban and issues monthly material purchasing plans to his suppliers. Once the finished parts are moved from the shipping area, assembly kanbans K_3 are removed from the container and sent to the assembly station which then produces the sun roof according to the K_3 kanban number. These kanbans will be attached to the container of the finished parts again.

In order to assemble the sun roof, the operator at the assembly station has to go to the WIP area of the forming station to withdraw semi-finished parts. The operator has to detach the K_2 kanban from the container, put it in a kanban box, and move the parts back to the assembly station.

Similarly, the forming station has to produce the parts according to the K_2 kanban number, and the operator has to go to the part supply area to withdraw the supply parts. The operator has to detach the K_1 kanban from the container, put it in a kanban box, and move the parts back to the forming station. The K_1 kanban will be sent back to the supplier that transports parts to the case company once a day according to the K_1 kanban number.

Currently the case company empirically sets K_1 to 4, K_2 to 4, K_3 to 3 and kanban size to 4. Simulation model of the assembly station of the sun roof production line is illustrated in Fig. 3. Simulation models of other stations are built in the sam way.

Two measures are used to determine the production performance of the JIT system: mean throughput rate (TR) and total work-in-progress (WIP). TR is defined as the ratio of the finished good parts to the total production time. TR is used to measure the production efficiency of a production system. WIP is defined as the total part number in the production line a day. WIP is used to measure the inventory cost of a production system.

3.2. Regression model

After the simulation model is built and its validity is verified, five levels of the kanban number of K_1, K_2, K_3, and the kanban size of B are tested. The levels are set at 2–6. Therefore, a 5^4 factorial experimental design is used, and five replications are tested for each experimental trial. Response surface method (RSM) is then used to build the regression model to create the relationships between the station kanban number, kanban size and the production performance for the sun roof simulation result. Since two performance measures are used in this research, two regression models have to be built. Linear and quadratic models are the common response models used to build the relationship between the kanban number, kanban size and production performance. The linear model will be applied first to fit the simulation result. If the fitness of the linear model is not satisfied, the quadratic model will be used. The linear regression models for TR and WIP are as follows:

$$Z_{TR} = \hat{\beta}_0 + \sum_{i=1}^{3} \hat{\beta}_i \cdot K_i + \hat{\beta}_4 \cdot B, \quad (2)$$

$$Z_{WIP} = \hat{\beta}_0 + \sum_{i=1}^{3} \hat{\beta}_i \cdot K_i + \hat{\beta}_4 \cdot B, \quad (3)$$

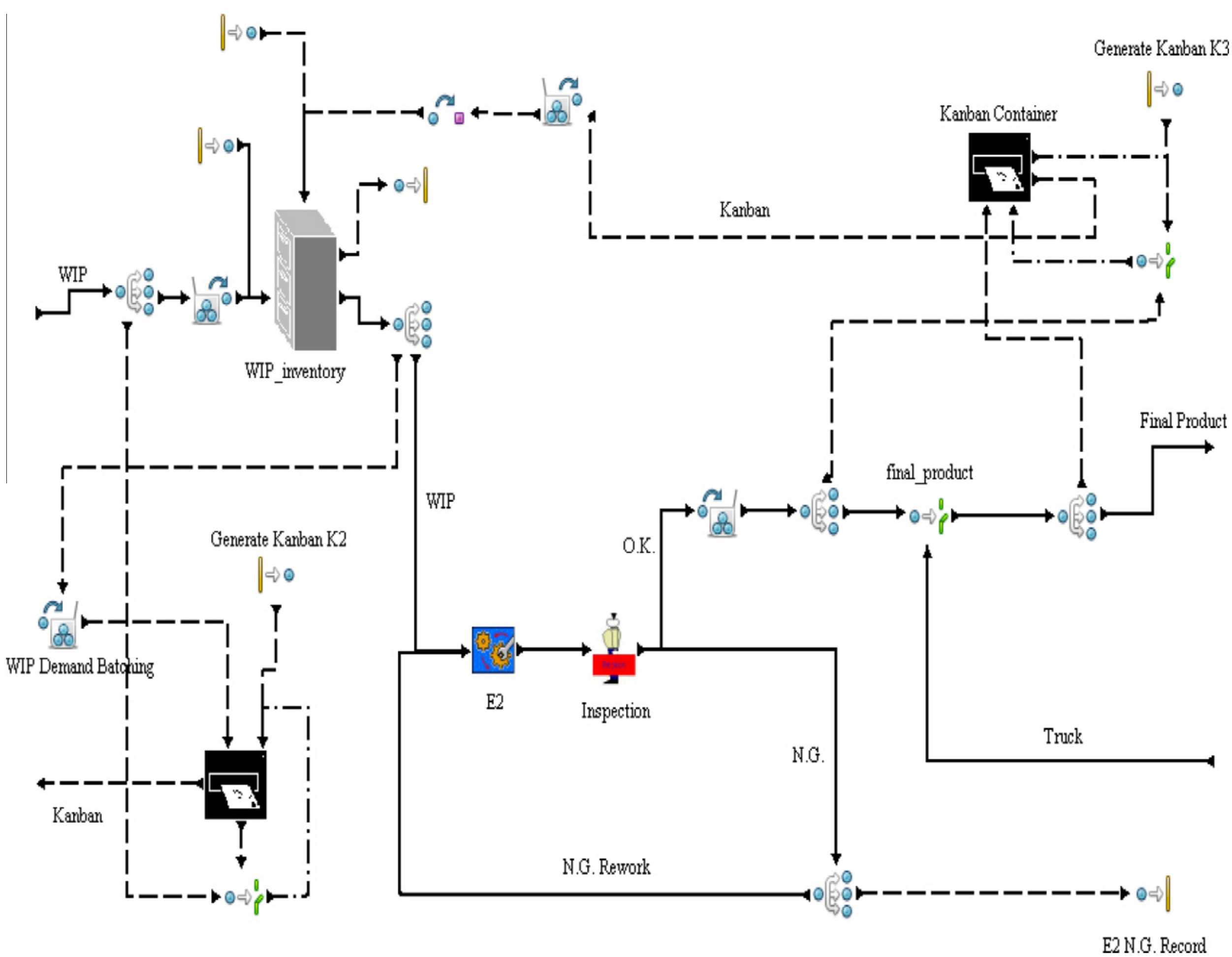

Fig. 3. Simulation model of the assembly station of the sun roof production line.

where, β_0 is a constant value, K_i is the kanban number, B is the kanban size, and $\hat{\beta}_i, \hat{\beta}_4$ are partial regression coefficient.

The quadratic regression models for TR and WIP are as follows:

$$Z_{TR} = \hat{\beta}_0 + \sum_{i=1}^{3} \hat{\beta}_i \cdot K_i + \hat{\beta}_4 B + \sum_{i=1}^{3} \hat{\beta}_{ii} K_i^2 + \hat{\beta}_{44} B^2 + \sum_{i=1}^{3}\sum_{j=1}^{3} \hat{\beta}_{ij} K_i K_j + \sum_{i=1}^{3} \hat{\beta}_{i4} B, \tag{4}$$

$$Z_{WIP} = \hat{\beta}_0 + \sum_{i=1}^{3} \hat{\beta}_i \cdot K_i + \hat{\beta}_4 B + \sum_{i=1}^{3} \hat{\beta}_{ii} K_i^2 + \hat{\beta}_{44} B^2 + \sum_{i=1}^{3}\sum_{j=1}^{3} \hat{\beta}_{ij} K_i K_j + \sum_{i=1}^{3} \hat{\beta}_{i4} B. \tag{5}$$

3.3. The proposed MOGA approach

Procedures of the proposed MOGA approach are as follows:

Step 1: chromosome encoding
A chromosome of the evolutionary algorithm is composed of the four decision variables K_1, K_2, K_3, B. The binary string representation for a chromosome is adopted and each decision variable is encoded into 4 binary digits. For example, for the case $K_1 = 4$, $K_2 = 4$, $K_3 = 3$, $B = 7$, the chromosome is coded as 0100010000110111.
Step 2: population initialization
Initial chromosomes are randomly generated, and the chromosome population size is denoted as N_{pop}. In the meantime, the elite set E and the temporary elite set E' are set to be empty.
Step 3: calculate the fitness value for each chromosome
The regression models of Z_{TR} and Z_{WIP} are used to calculate the objective values for each chromosome and the ***score(x)*** for each chromosome $\boldsymbol{x}$ is then calculated using Eq. (1). The ***score(x)*** value is generated using the GPSIFF that considers the quantitative fitness values in Pareto space for both dominated and non-dominated solutions. The non-dominated solutions will be copied to the temporary elite set E' for updating the elite set E.
Step 4: updating the elite set
First of all, the non-dominated solutions in the temporary elite set E' are put into the elite set E. The temporary elite set E' is then reset to be empty. Then the solutions in the elite set E are evaluated again for their non-dominance, and the dominated solutions are removed from the elite set. If the non-dominated solution number N_E is larger than the elite set capacity N_{max}, then some of the non-dominated solutions will be randomly selected to be removed from the elite set.
Step 5: selectionSelection is an operation to select chromosomes for generating the new offspring. A chromosome with a higher fitness has a higher probability of being selected for mating. The roulette wheel method is applied to the chromosome selection. In this research, $N_{pop} - N_{ps}$ number of chromosomes with high fitness are selected from the parent population, where $N_{ps} = N_{pop} \times Ps$, and Ps is a random number. If $N_{ps} > N_E$, then let $N_{ps} = N_E$, where N_E is the total chromosome number in the elite set E. In addition, N_{ps} elite chromosomes are randomly selected from the elite set E to form the new offspring population.
Step 6: crossoverFor each selected pair, a single-cut-point crossover operation is applied to generate an offspring with the crossover probability P_c. A random number R_c is generated. If $R_c \leqslant P_c$, then crossover operation is taken; otherwise, go to step 7. An example of the crossover operation is illustrated in Fig. 4.
Step 7: mutationIn this study, the one-gene mutation operation with a preset mutation probability P_m is applied to generate the new chromosomes. A random number R_m is generated. If $R_m \leqslant P_m$, then mutation operation is taken, and the bit with 1 is changed to 0, or 0 changed to 1. By using the crossover and mutation operations, a new offspring population is created. The parent chromosome populations are then replaced with the new offspring populations.
Step 8: termination test
A large generation number G_{max} is used as the stopping criteria in this study. If a pre-specified stopping condition is not satisfied, go to Step 3. Otherwise, the algorithm will stop and the Pareto-optimal front will be found from the population and the elite set E.

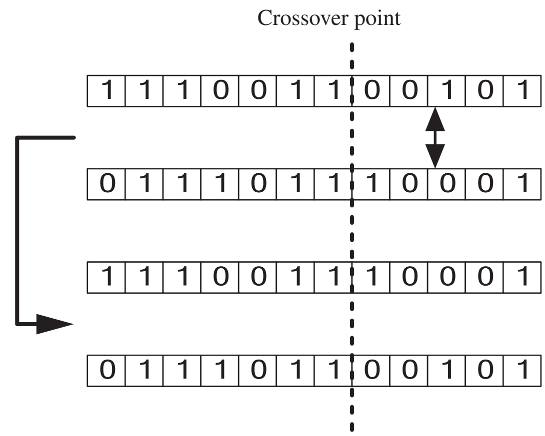

Fig. 4. Illustration of single-cut-point crossover operation.

4. Results

4.1. Results of simulation models and regression models

This research uses Simprocess to build the simulation model that consists of a supplier, a customer, and the case company. The simulation model is built based on the VSM as shown in Fig. 2. There are many modular in this simulation model. A simulation module of the assembly station is shown in Fig. 3. The real operation data of the case company is input to the simulation model. They are shown in the following:

(1) Monthly demands of the customer are 680 units.
(2) Monthly demands are leveled to each day and each delivery. There are three times of delivery each day, and the delivery interval is 2.5 h.
(3) The production cycle time (CT) of the forming station is 230 s, and the WIP has to be dried for 2 h before it can be used by the assembly station. The production CT of the assembly station is 550 s.
(4) Inferior rate for each station is assumed to be 1%.
(5) Machine utilization for each station is assumed to be 92%.
(6) There are 19 working days in a month, and 8 working hours a day.

This simulation model is run 19 days by using the kanban size $B = 4$, and kanban number $K_1 = 4$, $K_2 = 4$, and $K_3 = 3$ that are the kanban size and number settings of the case company. Production performance of WIP of the simulation is collected and compared with that of the real WIP of the case company. The Mann–Whitney U test of SPSS is conducted, and the P value is equal to 0.163 which shows that the WIP of the simulation model is insignificant difference from that of the real WIP of the case company. Therefore, this test verifies the validity of the simulation model. The simulation

Table 1
The Pareto-optimal solutions and their performance.

No.	Chromosome	(K_1, K_2, K_3, B)	TR	WIP	Improvement (TR ,WIP)
1	0010 0100 0100 0001	(2, 4, 4, 1)	15.947	3.958	(−66.7%, 87.9%)
2	0011 0111 0111 0001	(3, 7, 7, 1)	27.842	5.937	(−41.9%, 81.9%)
3	0100 1001 1001 0001	(4, 9, 9, 1)	35.842	7.917	(−25.2%, 75.9%)
4	0011 0101 0101 0010	(3, 5, 5, 2)	39.579	11.875	(−17.5%, 63.8%)
5	0011 0110 0110 0010	(3, 6, 6, 2)	45.684	12.056	(−4.7%, 63.2%)
6	0011 0111 0111 0010	(3, 7, 7, 2)	51.263	26.390	(6.9%, 19.5%)
7	0010 0100 0101 0100	(2, 4, 5, 4)	51.632	26.469	(7.7%, 19.3%)
8	0111 1000 1000 0010	(7, 8, 8, 2)	51.684	28.725	(7.8%, 12.4%)
9	0101 0110 0111 0011	(5, 6, 7, 3)	51.789	31.004	(8.0%, 5.5%)
10	0100 1000 1000 0100	(4, 8, 8, 4)	51.895	39.971	(8.2%, −21.9%)
	Current situation	(4, 4, 3, 4)	47.947	32.796	

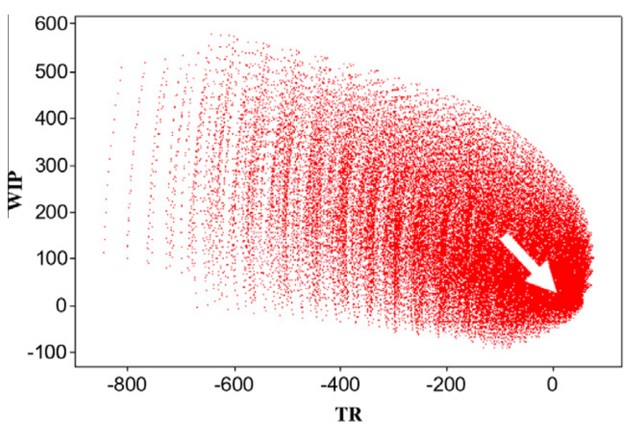

Fig. 5. TR and WIP performance space.

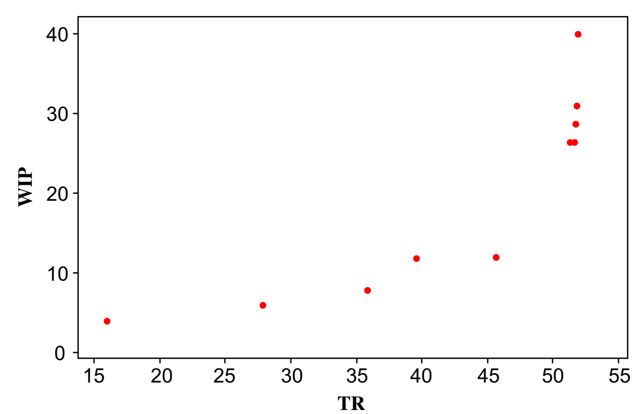

Fig. 6. The performance space of the 10 solutions.

model is then run by using the 5^4 factorial designs described in Section 3.2, and the system performance of TR and WIP are collected to build the regression models.

The linear model is applied first to fit the simulation result, but the fitness of the linear model is not satisfied because its R^2 value is too small. The quadratic models are then built by using central composite design (CCD) method. The quadratic regression models for TR and WIP are shown as follows:

$$\begin{aligned} Z_{TR} = & -124.714 + 20.595K_3 + 19.762K_2 + (-1.321)K_1 + 37.762B \\ & + 2.125K_2K_3 + (-1.75)K_3B + 0.25K_1K_2 + (-1.75)K_2B \\ & + (-0.125)K_1B + (-2.387)K_3^2 + (-2.387)K_2^2 + 0.113K_1^2 \\ & + (-2.262)B^2, \end{aligned}$$

$$\begin{aligned} Z_{WIP} = & -0.39026 + 0.35075K_3 + (-1.2689)K_2 + 1.97966K_1 \\ & + (-2.16586)B + (-0.36194)K_2K_3 + 0.23498K_1K_3 \\ & + 0.06542K_3B + 0.02560K_1K_2 + 0.74785K_2B \\ & + 1.55637K_1B + (-0.05249)K_3^2 + 0.22577K_2^2 \\ & + (-0.27467)K_1^2 + 0.17302B^2. \end{aligned}$$

The R^2 values for the TR and WIP models are 0.919 and 0.998 respectively. These models can be used to generate the TR and WIP performance for all the possible combinations of K_1, K_2, K_3, and B whose values fall in the interval 0–15. The TR and WIP performance space is shown in Fig. 5. The arrow in the figure shows the Pareto-front solutions.

4.2. Results of the MOGA approach

There are many parameters in this MOGA approach, including generation number G_{max}, population size N_{pop}, the elite set capacity N_{max}, crossover rate P_c, and mutation rate P_m. Different parameter combinations may result in different solutions. An L_{27} orthogonal array experiment is conducted to find the best parameter combination regarding to solution error ratio (ER). In this experiment, the generation number is set at 50, 100, 150; the population size at 30, 60, 90; the elite set capacity at 3, 6, 9; the crossover rate at 0.6, 0.7, 0.8; and the mutation rate at 0.005, 0.01, 0.015. Experimental results show that the combination of (150, 30, 6, 0.7, 0.01) has the best solution. This parameter combination is then used in the MOGA approach to find the Pareto-front solutions in Fig. 5.

The Pareto-optimal solutions found by MOGA should be feasible in practice. Therefore, the solutions that result in zero TR value or have a zero value in kanban number or kanban size are deleted from the candidate solution pool. The final 10 solutions, their original chromosomes, TR value, WIP value, and improvement in TR and WIP comparing with that of the current situation are shown in Table 1. This table does show the trade-off between TR and WIP in the solutions. The solutions 6–9 have better performance in both TR and WIP than the current kanban number and kanban size used in the case company. The performance space of the 10 solutions are shown in Fig. 6 which demonstrates that the 10 solutions do construct the Pareto-front.

5. Conclusions

Kanban is a tool used to achieve JIT production. The container size is equivalent to the kanban size and the kanban number represents the WIP container number. A large kanban size and kanban number means a high WIP level. Although it is a traditional problem to determine the kanban number and kanban size, none a research determines them simultaneously. However, there is a trade-off between the kanban size and kanban number. When the kanban size is larger, the kanban number becomes smaller. It is necessary to determine the kanban size and kanban number simultaneously.

In addition, different kanban number used at different operation stations may result in different production performance in WIP and throughput rate. The kanban number used at each station has to be determined by considering multi-objectives. Therefore, the objective of this research is to propose an integrated multi-objective genetic algorithm based system to determine the optimal kanban size and kanban number simultaneously for a JIT system.

In the proposed integrated system, a simulation model is built to simulate the multi-stage JIT production system of the company. Then an experimental design of different kanban numbers and sizes for different production stages is applied to test the production performances. Based on the experimental design and simulation results, regression models are built to represent the relationships between the kanban numbers and kanban size of different production stages and the production performance. These regression models are then used in genetic algorithms to generate the performance for chromosomes. Finally, the proposed MOGA based system uses the GPSIFF as the fitness function to evaluate the multiple objectives for chromosomes and used to find the Pareto-optimal kanban number and size for multiple objectives.

This research finds that some of the Pareto-optimal solutions have better performance in both TR and WIP than the current kanban number and kanban size used in the case company. This result demonstrates the feasibility of the proposed integrated system. Although the proposed system is mainly for determining the kanban number and size for a traditional kanban system, it may be applied to a flexible kanban system. The extension of the proposed system to a flexible kanban system may be investigated in the future research.

Acknowledgement

This research was supported in part by the National Science Council of the Republic of China under grant number NSC 97-2221-E-224-031.

References

Chan, F. T. S. (2001). Effect of kanban size on just-in-time manufacturing systems. *Journal of Materials Processing Technology, 116*, 146–160.

Deb, K. (1999). Multi-objective genetic algorithms problem-difficulties and construction of test problems. *Evolutionary Computation Journal, 7*(3), 205–230.

Duri, C., Frein, Y., & Di Mascolo, M. (1995). Performance evaluation of kanban multiple-product production systems. *Emerging Technologies and Factory Automation, 3*, 557–566.

Guneri, A. F., Kuzu, A., & Taskin Gumus, A. (2008). Flexible kanbans to enhance volume flexibility in a JIT environment: a simulation based comparison via ANNs. *International Journal of Production research*(September). 15.

Gupta, S. M., & Al-Turki, Y. A. (1997). An algorithm to dynamically adjust the number of kanbans in stochastic processing times and variable demand environment. *Production Planning & Control, 8*(2), 133–141.

Ho, S. Y., Shu, L. S., & Chen, J. H. (2004). Intelligent evolutionary algorithms for large parameter optimization problems. *IEEE Transactions Evolutionary Computation, 8*(6), 522–541.

Köchel, P., & Nieländer, U. (2002). Kanban optimization by simulation and evolution. *Production Planning and Control, 13*(8), 725–734.

Lee, I. (2007). Evaluating artificial intelligence heuristics for a flexible kanban system: Simultaneous kanban controlling and scheduling. *International Journal of Production Research, 45*(13), 2859–2873.

Mansouri, S. A. (2005). A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *European Journal of Operation Research, 167*, 696–716.

Monden, Y. (1993). *Toyota production system: An integrated approach to just-in-time* (2nd ed.). Industrial Engineering and Management Press, pp. 279–290.

Nori, V. S., & Sarker, B. R. (1998). Optimum number of kanbans between two adjacent stations. *Production Planning and Control, 9*(1), 60–65.

Sivakumar, G. D., & Shahabudeen, P. (2008). Algorithms for the design of a multi-stage adaptive kanban system. *International Journal of Production Research*(September). 11.

Savsar, M., & Al-Jawini, A. (1995). Simulation analysis of just-in-time production systems. *International Journal of Production Economics, 42*, 48–67.

Yavuz, I. H., & Stair, A. (1995). A kanban-based simulation study of a mixed model just-in-time manufacturing line. *International Journal of Production research, 33*(4), 1027–1048.

Wang, S., & Sarker, B. R. (2006). Optimal models for a multi-stage supply cha- in system controlled by kanban under just-in-time philosophy. *European Journal of Operational Research, 172*, 179–200.