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 © 2010 Elsevier Ltd. All rights reserved.
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