International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Volume7, Issue1 (March 2022 - May 2022) | Pages 77-89

# DESIGN OF SINGLE STAGE ADAPTIVE KANBAN SYSTEM USING MOGA APPROACH

Karuppasamy P, Ramachandran V, Sudhan A
Department of MechanicalEngineering,
SethuInstituteofTechnology, Pulloor, Kariapatti, India

## ABSTRACT

Aims to designing production control scheme for Kanban-based just-in-time (JIT) environment. The traditional kanban system with fixed number of cards does not work satisfactorily in unstable environment. In the adaptive kanban-type pull control mechanism the number of kanban is allowed to change with respect to the inventory and backorder level. It is required to set the threshold values at which cards are added or deleted which is a part of the design. Previous studies used the GA to design the adaptive kanban system. In this paper Multi Objective Genetic Algorithm (MOGA) heuristics are developed and used to set the design parameters of adaptive kanban system. The numerical results have been compared with GA and objective function and number of cards should be optimized using Multi Objective Genetic Algorithm (MOGA).

## 1.0 INTRODUCTION

Production is the process of converting the input resources like men, material and money into a specified set of output elements like products or services. Production systems form the base of any organized factory. It involves planning, execution and control to carry out production process. With the consistent need to reduce manufacturing cost and to meet global competition, some new concepts, which are better than traditional production systems, began to evolve and attain importance. Just-In-Time (JIT) production system is one such concept.

## KANBAN SYSTEM

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. Kanban is a visual signal that's used to trigger an action. The word kanban is Japanese. Roughly translated, it means "card you can see."

Toyota introduced and refined the use of kanban in a relay system to standardize the flow of parts in their production lines in the 1950s. Kanban was one of several tools Toyota developed to ensure that inventory was based on actual customer orders rather than managerial forecasts.

Kanban starts with the customer's order and follows production downstream. Because all requests for parts are pulled from the order, kanban is sometimes referred to as a "pull" system. At its simplest, kanban is a card with an inventory number that's attached to a part. Right before the part is installed, the kanban card is detached and sent up the supply chain as a request for another part. A part is only manufactured (or ordered) if there is a kanban card for it.

Motivated by the idea of readjusting the parameters of the system as it operates, this research aims at, developing a methodology for an adaptive kanban system for single stage production system where the number of cards in the system is dynamically readjusted based on current inventory and back order levels. The performance of a kanban system is the collective effect of various parameters such as number of kanbans, lot size, sequence rules, number of machines etc. (Berkley 1992). The estimation of these parameters gets complicated due to issues like variation in demand, variation in processing times, different types of products etc. The combinatorial property of such problems warrants development of some efficient methodology or heuristic to obtain a good solution. Hence this research also aims at exploring the possibility of applying meta heuristic techniques in the design of AKS.

## METHODOLOGY

### ADAPTIVE KANBAN SYSTEM (AKS)

Traditional Kanban System (TKS) is successful in a production environment with stable demand and lead time. However, when there is a wide variation in demand, with fixed number of cards the manufacturing process experiences either excess inventory or shortages. To overcome this AKS is proposed in which, the number of cards can be varied. A detailed study of AKS for single stage presented in this chapter.

### SINGLE STAGE ADAPTIVE KANBAN SYSTEM

Tardif and Maaseidvaag (2001) devised AKS to handle the variable supply and demand condition. Here the number of cards in use is allowed to vary, based on the current inventory and backorders. An extra card is added to the system if a demand arrives while the inventory level is below a release threshold (R). When the inventory level exceeds a capture threshold (C) a card is retrieved from the system.Queue P contains finished parts, and queue D contains backordered demands. The single stage adaptive mechanism uses K kanban cards and E extra cards. Again, MP represents the manufacturing process. Initially, before any customer demands arrive at the system, P has a base stock of K finished parts, queue E contains all the extra cards, and D and MP are empty. Let N (t) be the total number of parts in queue P minus the total number of backorders in queue D at time t. Let X (t) be the number of extra cards not in use at time t.

Let R be the release threshold when one adds extra card and C be the capture threshold when one retrieves an extra card from circulation. Upon the arrival of a customer demand at time t but before the part is released to the customer, if N (t)≤R and X (t)>0, an extra card is immediately released and sent to MP. Then, a part immediately releases its card and is given to the customer. However, if the value of N (t) before the release of the part is greater than C, then the card released with a part is not sent to MP but is recaptured and stored in E.

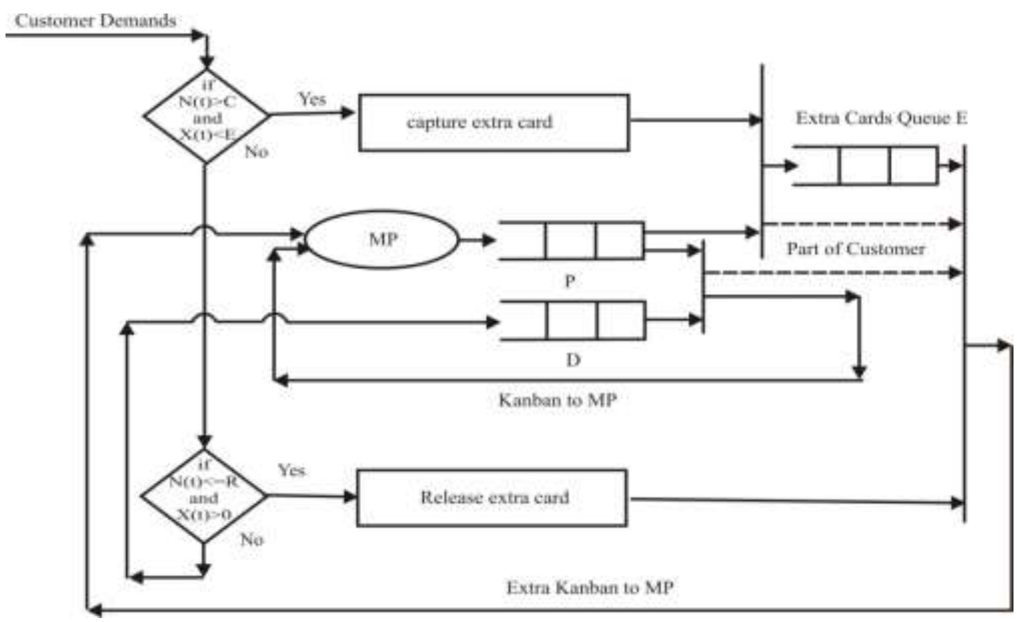


Figure 1 Model for a single stage adaptive kanban system

Single-stage adaptive kanban mechanism described above. The dotted line represents the movement of material whereas continuous lines represent the movement of information and kanban cards. The thick vertical line represents the synchronization of material and information.

The system is modeled in terms of the state (i,x) whose evaluation describes a Continuous Time Markov Chain (CTMC). State (i,x) denotes the total number of parts in queue P minus total number of back orders in queue D with i, and the number of extra cards in circulation with x.

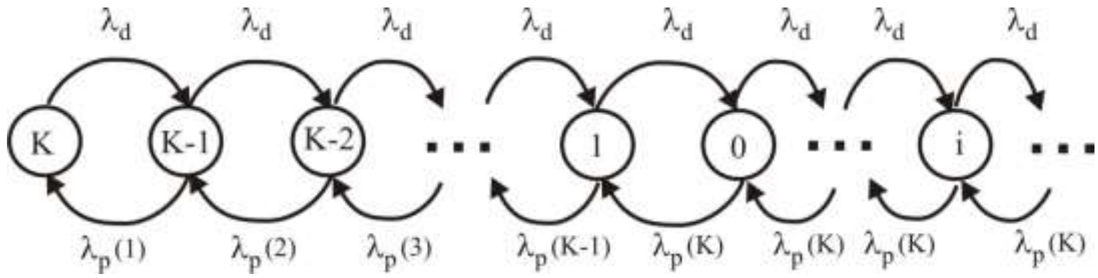


Figure 2 State - diagram representing the behavior of the system

The above diagram in Markovian model of the AKS system when R>0. Here R must be less than C and C must be less than or equal to K+1 to ensure that the system be able to return to the initial state (K,0).

Here when

$\lambda_d / \lambda_p(K+E) < 1$

the steady state probabilities exist.

- An attempt is made to estimate the parameters required for the design of AKS using the following heuristics
- Genetic Algorithm
- Multi objective Genetic Algorithm (MOGA)

The description of the algorithms and the design of the experiments based on these heuristics are Discussed.

The detailed study of the adaptive kanban system for single stage. The CTMC model for the adaptive kanban system is presented.

## EXPERIMENTAL ANALYSIS

## MULTI-OBJECTIVE GENETIC ALGORITHMS (MOGA) BASED SEARCH HEURSITIC

Being a population based approach, MOGA are well suited to solve multi-objective optimization problems. A generic Multi-objective GA can be easily modified to find a set of multiple non-dominated solutions in a single run. The ability of MOGA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces. The crossover operator of GA may exploit structures of good solutions with respect to different objectives to create new non-dominated solutions in unexplored parts of the Pareto front. In addition, most multi-objective GA do not require the user to prioritize, scale, or weigh objectives. Therefore, MOGA has been the most popular heuristic approach to multi-objective design and optimization problems. Jones et al. reported that 90% of the approaches to multiobjective optimization aimed to approximate the true Pareto front for the underlying problem. A majority of these used a meta-heuristic technique, and 70% of all meta-heuristics approaches were based on evolutionary approaches.

Multi-objective optimization deals with solving optimization problems which involve multiple objectives. Most real-world search and optimization problems involve multiple objectives (such as minimizing fabrication cost and maximize product reliability and others) and should be ideally formulated and solved as a multi-objective optimization problem. However, the task of multi-objective optimization is different from that of single-objective optimization in that in multi-objective optimization, there is usually no single solution which is optimum with respect to all objectives. The resulting problem usually has a set of optimal solutions, known as Pareto-optimal solutions, non-inferior solutions, or effective solutions (Steuer, 1986). Since there exists more than one optimal solution and since without further information no one solution can be said to be better than any other Pareto-optimal solution, one of the goals of multi-objective optimization is to find as many Pareto-optimal(Pareto-front) solutions as possible. Therefore different solutions will produce trade-off between different objectives and a set of solutions is required to represent the optimal solutions of all objectives.

## MULTI-OBJECTIVE OPTIMIZATION FORMULATION

In this work, is to minimize two objectives like inventory (both finished and semi finished goods) and back orders, each of them having the four decision variable like K,E,R,C.

First in this method operates with a collection of chromosomes, called a population. The population is normally randomly initialized. Chromosomes are made of discrete units called genes. Each gene controls one or more features of the chromosomes. After that find the Pareto front of each gens like K,E,R,C. Normally Pareto front of the each gens lies in the some minimum and maximum range, that ranges are identifies with respect to the two different objective function. And then these ranges are applied in the existing population. Finally obtained new population from the Pareto approach. In figure 4.2 show the flow chart proposed adaptive kanban system (AKS).

In this way the entire solution space may be explored and multiple solutions detected. Evaluation of the individuals in the population is accomplished by calculating the objective function value of the problem using the parameter set. The result of the objective function calculation is used to calculate the fitness value of the individual. Fitter chromosomes have higher probabilities of being selected for the next generation. After several generations, the algorithm converges to the best chromosome, which hopefully represents the optimum or near optimal solution.

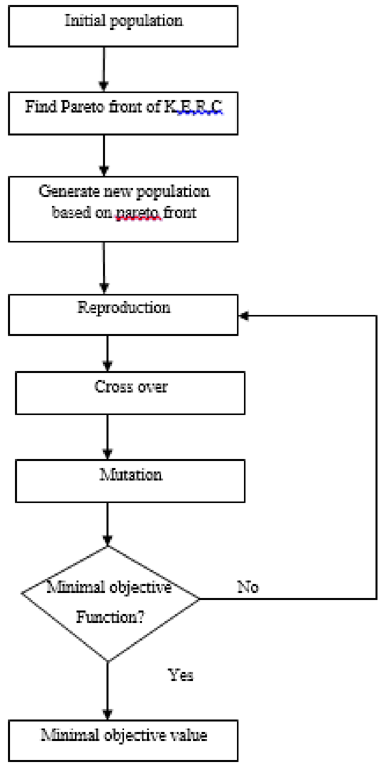


Figure 3 Flow Chart of Proposed Methodology

The mutation operator introduces random changes into characteristics of chromosomes. Mutation is generally applied at the gene level. In typical MOGA implementations, the mutation rate (probability of changing the properties of a gene) is very small, typically less than 1%. Therefore, the new chromosome produced by mutation will not be very different from the original one. Mutation plays a critical role in MOGA. The crossover leads the population to converge by making the chromosomes in the population alike. Mutation reintroduces genetic diversity back into the population and assists the search escape from local optima. Finally obtain the minimal objective function values. The goal of the multi objective genetic algorithm is to find a set of solutions in that range (ideally with a good spread). The set of solutions is also known as a Pareto front. All solutions on the Pareto front are optimal.

## SETTING POPULATION SIZE AND NUMBER OF GENERATION

In MOGA population size and number of generation are the parameter related to the convergence of algorithm. The experiment is conducted varying population size and the convergences of the problem with respect to different number of generations are recorded as shown in Figure.

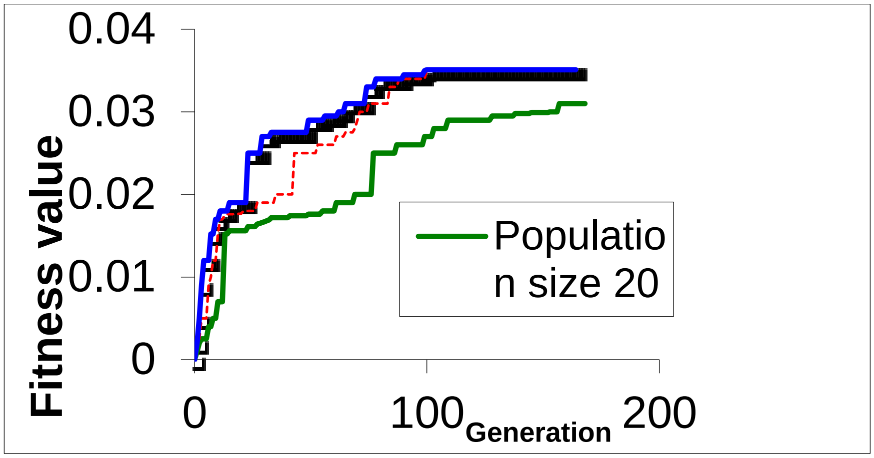


Figure 4 GA convergence for single stage AKS

Compared to population size 20 the other two population sizes namely 30and 50 give better convergence. However, as the number of generations reaches 100, the results obtained with both population size 30 and 50, merge together. Hence to reduce the CPU time population size is considered as 30 and number of generation is fixed as10.

## RESULTS AND DISCUSSION

MOGA based search models developed to estimate the near optimal parameters of AKS are given in the previous chapters. A similar model is developed for GA and used as the base for comparison. The models are implemented using MATLAB and tested with several examples. The details of the cases used for the numerical experiments for single stage AKS models, and the comparison of performance measures are discussed

## SINGLE STAGE ADAPTIVE KANBAN SYSTEM

In an single stage adaptive kanban system numerical experiment are solved in following

## NUMERICAL EXPERIMENTS

Case 1: Consider a MP consisting of a single workstation. The demand follows the Poisson process with rate $\lambda d$ =0.2. The MP consists of 5 parallel servers. The processing times are exponentially distributed with mean 7. Therefore average service rate of the MP is given by

$$\lambda_p(n) = \begin{cases} n/7, & n < 5, \\ 5/7, & n \geq 5 \end{cases}$$

The back order penalty cost b is assumed to be 100.

The details of other cases used for numerical experiment are given in Table 1. In all the cases the processing times are exponentially distributed and the demand follows Poisson process.

Table 1 The performance of GA and the design of AKS is verified using these 12 cases

| Case No | Type | No .of. m/c's | Processing Time mean | Mean Rate (μ) | Demand Mean | Demand Rate (λd) | Cost ratio (b) |
|---|---|---|---|---|---|---|---|
| 1 | Parallel | 5 | 7 | 0.1429 | 5 | 0.2000 | 100 |
| 2 | Parallel | 10 | 6 | 0.1667 | 7 | 0.1429 | 500 |
| 3 | parallel | 10 | 20 | 0.0500 | 15 | 0.0670 | 700 |
| 4 | Parallel | 10 | 6 | 0.1667 | 5 | 0.2000 | 1000 |
| 5 | Parallel | 15 | 9 | 0.1110 | 7 | 0.1429 | 500 |
| 6 | tandem | 4 | 5 | 0.2000 | 10 | 0.1000 | 100 |
| 7 | tandem | 4 | 5 | 0.2000 | 8 | 0.1250 | 100 |
| 8 | tandem | 6 | 5 | 0.2000 | 8 | 0.1250 | 100 |
| 9 | tandem | 7 | 9 | 0.1110 | 20 | 0.0500 | 500 |
| 10 | tandem | 10 | 3 | 0.3333 | 7 | 0.1429 | 400 |
| 11 | tandem | 10 | 3 | 0.3333 | 10 | 0.1000 | 500 |
| 12 | tandem | 11 | 12 | 0.0830 | 40 | 0.0250 | 400 |

Table 2 Result of the numerical experiment

| Case | GA | | MOGA | |
|---|---|---|---|---|
| | Cards (K,E,R,C) | Z | Cards (K,E,R,C) | Z |
| 1 | 4,2,2,3 | 5.4703 | 3,3,3,4 | 5.4360 |
| 2 | 3,12,3,4 | 4.7002 | 4,5,2,3 | 4.5354 |
| 3 | 5,13,2,5 | 5.8334 | 5,4,2,3 | 5.8152 |
| 4 | 4,6,3,4 | 5.6122 | 5,5,2,3 | 5.6001 |
| 5 | 5,11,2,3 | 5.7214 | 5,6,2,3 | 5.5508 |
| 6 | 10,2,10,11 | 13.9561 | 12,3,5,6 | 13.8205 |
| 7 | 18,13,5,6 | 21.4600 | 19,11,3,6 | 21.05 |
| 8 | 20,15,10,11 | 27.4700 | 23,13,7,8 | 26.9476 |
| 9 | 18,20,6,7 | 19.1187 | 16,20,6,7 | 18.5929 |

| | | | | |
|---|---|---|---|---|
| 10 | 17,10,5,11 | 21.4603 | 18,9,7,8 | 208882 |
| 11 | 13,14,3,7 | 14.0700 | 12,10,5,6 | 13.7186 |
| 12 | 12,19,4,6 | 14.2370 | 13,19,3,6 | 13.8969 |

The performance of MOGA based AKS models is analyzed under the following two categories

- Objective function value (Z)
- Number of cards used

## RESULTS AND DISCUSSION

The objective of the model is to minimize the cost. Improvement in Z value while using MOGA for AKS is compared with the GA.

While using GA the improvement in Z ranges 0.63% to 3.63% over GA. The average improvement in Z value is 1.95% over GA. From the Table it is observed that for all the cases, the AKS parameter of MOGA improves when compared with AKS parameters of GA. The individual percentage improvement in Z using MOGA for AKS over the GA are shown in Table.

Table 3 Improvement in Z using MOGA single stage AKS

| Case | Improvement in Z using MOGA (%) |
|---|---|
| | Over GA Method |
| 1 | 0.63 |
| 2 | 3.63 |
| 3 | 0.31 |
| 4 | 0.22 |
| 5 | 3.07 |
| 6 | 0.98 |
| 7 | 1.95 |

| | |
|---|---|
| 8 | 1.94 |
| 9 | 2.83 |
| 10 | 2.74 |
| 11 | 2.56 |
| 12 | 2.45 |
| Average Improvement | 1.95 |

**CARD USED**

In TKS a fixed number of cards are used, while in AKS the number of cards in use ranges from K to K+E . It is found that for almost all the cases the number of cards used in MOGA should be lower than that of GA due to this minimize the cost of inventory.

Table 4 Comparison of Card used in AKS

| Case | AKS (GA) | | | AKS (MOGA) | | |
|---|---|---|---|---|---|---|
| | Cards Used | | | Card used | | |
| | Min | Max | Average | Min | Max | Average |
| 1 | 4 | 6 | 5.0 | 3 | 6 | 4.5 |
| 2 | 3 | 15 | 9.0 | 4 | 9 | 6.5 |
| 3 | 5 | 18 | 11.5 | 5 | 9 | 7 |
| 4 | 4 | 10 | 7.0 | 5 | 10 | 7.5 |
| 5 | 5 | 16 | 10.5 | 5 | 11 | 8 |
| 6 | 10 | 12 | 11.0 | 12 | 15 | 13.5 |
| 7 | 18 | 31 | 24.5 | 19 | 30 | 24.5 |
| 8 | 20 | 35 | 27.5 | 23 | 36 | 29.5 |
| 9 | 18 | 38 | 23.0 | 16 | 36 | 26 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | 17 | 27 | 17.0 | 18 | 27 | 22.5 |
| 11 | 13 | 27 | 20.0 | 12 | 22 | 17 |
| 12 | 12 | 31 | 21.5 | 13 | 32 | 22.5 |

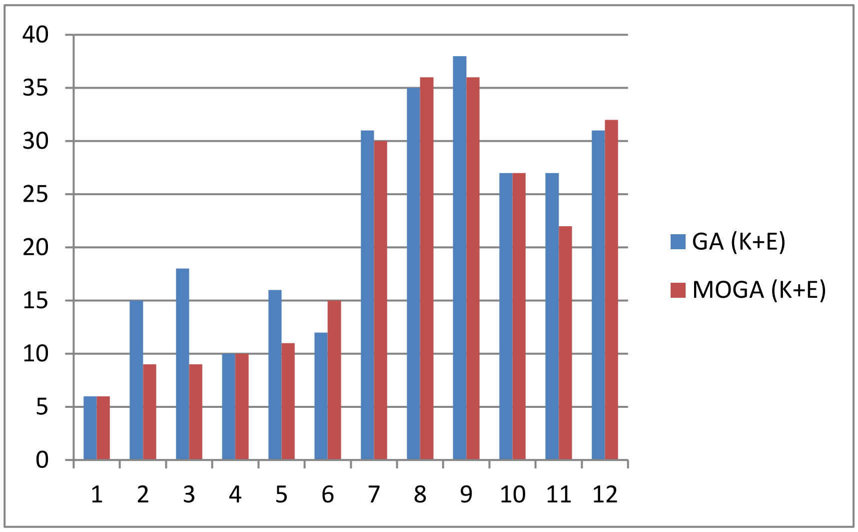


Figure 5 Comparison of Card used in AKS

Except case 6,8 and 12,the remaining cases is found that the upper limit of cards used in AKS by MOGA itself is less than that of GA, comparision of card uses are shown in figure.

Hence by adopting AKS better Z is obtained using less number of cards than equivalent GA .The proposed system can find the optimal Objective function value (Z) and number of cards for a single-stage adaptive kanban system.

## CONCLUSIONS

Kanban is a tool used to achieve JIT production. Traditional kanban system with fixed number of kanban performs unsatisfactorily under unstable demand. Adaptive kanban is proposed to overcome this problem. This research finds that some of the Important-optimal solutions have better performance over the GA. While using MOGA based AKS improvement in Z ranges 0.63% to 3.63% over GA. The average improvement in Z value is 1.95% over GA. AKS allows flexibility of varying number of cards used based on certain threshold values. In comparing with GA, Multi objective genetic algorithm (MOGA) will reduce the card usage, which results that inventory should be minimized so consequent of this effective use of resource.

Since that objective is to be minimize the cost associated with the backorder and the inventory.

**References:**

a) 1. *Di Mascolo, M., Frein, Y., & Dallery, Y. "An analytical method for performance evaluation of a kanban controlled production" , 1996.*
b) *2. Hall, R. W, "Zero inventories"(Irwin/Apics, series in production management). New York: McGraw-Hill Publications, 1983.*
c) *3. Yavuz, I. H., & Stair, A A kanban-based simulation study of a mixed model just-in-time manufacturing line. International Journal of Production research, volume 33(4), pages 1027–1048, 1995.*
d) *4 .Nori, V. S., & Sarker, B. R. "Optimum number of kanbans between two*
e) *adjacent stations". Production Planning and Control, volume 9(1), 60–65, 1998.*
f) *5 .Reeves, C. R. "Genetic algorithms, in modern heuristic techniques for combinatorial problems". New York: John Wiley and Sons Publication,1993.*
g) *6. Shahabudeen, P., Krishnaiah, K., & Gopinath, R. "Design of bi-criteria kanban system using simulated annealing technique". Computers and Industrial Engineering,*
h) *pages 355–370,2002.*
i) *7. Wang, S., & Sarker, B. R. "Optimal models for a multi-stage supply cha- in system controlled by kanban under just-in-time philosophy". European Journal of Operational Research, pages 179–200, 2006..*
j) *8.S. Afshin Mansouri. "A Multi-Objective Genetic Algorithm for mixed-model sequencing on JIT assembly lines", 2004*
k) *9. Shahabudeen.P & Sivakumar.G.D 'Design of multi-stage adaptive kanban system' Springer-verlog limited London 2007.*
l) *10. Tung-Hsu (Tony) Hou & Wei-Chung Hu. "An integrated MOGA approach to determine the Pareto-optimal kanban number and size for a JIT system",2010*
m) *11. N.Viswandham.,&Y.Narahari "Performance modeling of automated manufacturing system book".Prentice-Hall of india private limited, 2005*
n) *12. Schonberger 'Japanese production management an evalution with mixed success',Journal of operation management 25 (2007),403-419*
o) *13. Savsar, M. and Choueiki, M.H., 2000. A Neural network procedure for kanban allocation in JIT production control system. International Journal of Production Research, 38 (14), 3247–3265.*
p) *14. Sivakumar.G.D & Shahabudeen.P. 'Algorithm for the design of a Multi stage adaptive kanban system', tailor and francis group 2009, 6707–6738.*
q) *15. Maaseidvaag 'An adaptive approach to controlling kanban systems. European Journal of Operational Research 132(2) 411-424 (2001)*
r) *16. Berkly BJ. 'A review of kanban production control research literature'. Product Operation Management 1992; I: 393–411.*
s) *17. Sarker BR,& Balan CV. 'Operations planning for a single-stage kanban system operating under linear demand'. Int J Product Res 1998; 36(2):357–75.*
t) *18. Spearman ML, Zazanis MA. 'Push and pull production systems – issues and comparisons'. Oper Res 1992; 40(3):521–32.*

u) *19. Tardif, Valerie, & Maaseidvaag, Lars (2001). 'An adaptive approach to controlling kanban systems'. European Journal of Operation Research, 132(2), 411–424.*

v) *20. Shahabudeen.P&Sivakumar. G.D 'Algorithm for the design of single-stage adaptive kanban system', Computers & Industrial Engineering 54 (2008) 800–820.*

w) *21. Saravanan.R, 'Optimization of operating parameters for CNC manufacturing systems using Conventional and non-conventional techniques (GA)', PhD Thesis, Regional Engineering College, Bharathidasan University, Tiruchirappalli, 2001.*

x) *22. Gupta, S.M. and Al-Turki, Y.A.Y., 1997. An algorithm to dynamically adjust the number of kanbans in stochastic processing times and variable demand environment. Production Planning and Control, 8 (2), 133–141.*

y) *23. Paris, J.L., Tautou-Guillaume, L., and Pierreval, H., 2001. Dealing with design options in the optimisation of manufacturing systems: an evolutionary approach. International Journal of Production Research, 39 (6), 1081–1094.*

z) *24. Carlos Coello Coello. Recent trends in evolutionary multiobjective optimization. In Lakhmi Jain, Xindong Wu, Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing, pages 7–32. Springer Berlin Heidelberg, 2005. 10.1007/1-84628-137-72:*