Australian Journal of Basic and Applied Sciences, 4(3): 370-378, 2010
ISSN 1991-8178

Optimal Model for Supply Chain System Controlled by Kanban under JIT Philosophy by Integration of Computer Simulation and Genetic Algorithm

A. Azadeh, J. Layegh, P. Pourankooh

[1]Department of Industrial Engineering and [2,3]M. Saberi Center of Excellence for Intelligent Based Experimental Mechanics, College of Engineering, P.O. Box 11155-4563, University of Tehran, Iran
[2]Department of Industrial Engineering, University of Tafresh, Iran
[3]Institute for Digital Ecosystems & Business Intelligence, Curtin University of Technology, Perth, Australia

Abstract: In this study, multi-stage supply chain system (SCS) controlled by kanban system is evaluated. In the kanban system, decision making is based on determination of batch size for each kanban. This paper attempts to simulate supply chain system regarding the costs under just-in-time (JIT) production philosophy. Since the adopted model is of backward type, the desired output is given in order to find the parameters and/or the structure of the model producing the output. This backward problem is non-analytic and often seems to be even more complex than the forward one**.** The paper applies Genetic Algorithm (GA) to optimize this simulation model. A simple real-coded GA is presented and used to change the simulation model parameters. With each new set of parameters, a simulation run is performed. From the statistics gathered by running the simulation, a goal function is constructed to measure the quality of these parameters. GA successfully provides a parameter set to demonstrate its capability to solve such difficult backward problems even in the area of complex simulation model optimization specially when there is no prior knowledge of simulation model behavior. Significance: Since supply chain management has drawn much attention in industrial and academic fields, various techniques are developed to model, analyze and solve complex decision making problems in supply chains. Computer-based simulation with its own strength on evaluating variations and interdependencies in a complex system is one of those promising techniques. This paper addresses the successful application of GA-simulation, which is a random search technique, to simulation model optimization and design, though the stochastic behavior of SCS. Hence, we have used kanban to achieve SCS control goals under JIT philosophy. Moreover, an optimization algorithm is coupled to the model that directly changes control parameters of the model to increase its performance.

Key words: Supply chain system; Just-in-time; Simulation; Optimization; Genetic Algorithm (GA)

INTRODUCTION

Simulation is one of the most useful modeling tools to design many types of systems. Simulation models can incorporate a greater level of details and capture specific features of real objects, such as time dynamics and overall behavior. However, applying simulation models only for descriptive purposes does not alone justify the effort to build them. After building the model, which is hard enough, the designer uses the model to answer many ‘‘what-if ’’ questions, for example: “what will happen to the model behavior if some parameters change?”. The goal of these changes is to outperform some nominal designs, or even to build an entirely new one. Simulation Models (SMs) have specific features that make the application of classical optimization methods difficult or even impossible. In this paper, we demonstrate the successful application of so-called Genetic Algorithm (GA) to SM optimization and design, a problem considered to be intricate for classical optimization techniques (Andradottir, 1992; Evans *et al.,* 1991; Stuckman *et al.,* 1991; Tomkins and Azadivar, 1995). SM optimization and design have broad applicability since the analyst is often confronted with a variety of parameters which in total offer a large combinatorial problem to explore.

Corresponding Author: A. Azadeh, Department of Industrial Engineering and Center of Excellence for Intelligent Based Experimental Mechanics, College of Engineering, P.O. Box 11155-4563, University of Tehran, Iran
E-mail: aazadeh@ut.ac.ir or ali@azadeh.com

A supply chain system (SCS) is usually composed of a series of organizations and/or independent companies. This system is a set of approaches applied to an effective relationship between suppliers, manufacturers, warehouses, distribution centers, retailers and customers to produce and distribute in right quantity, to right location and at right time, in order to minimize the total system costs while satisfying the service level requirements (Shajun Wang, 2004). Long term profit that guaranties SCS perpetuity applies three traits as hereunder:

1) High quality production
2) Low cost
3) Right time delivery

In SCS, proper and on time delivery means that the goods should be delivered exactly regarding ordered quantity to right location at right time. Generally there are two types of SCS:

a) Single-stage supply chain system (SSSCS)
b) Multi- stage supply chain system (MSSCS)

If SCS consists of only two plants, it is called SSSCS, while MSSCS comprises more than two plants. In this paper, we use kanban to achieve SCS control goals. Kanban is a Japanese word for card and it is generally an information tool to develop information flow. Generally, there are two types of kanban: withdrawal kanban and production kanban. Kanban system is so efficient and easy to be implemented. In this system, each plant sends signals to preceding plant for needed parts and kanban system acts by considering customer demand at last plant. The plants located along the production lines only produce or deliver desired components when they receive a card. Empty containers show that more parts will be needed for production. Each workstation only produces enough components to fill containers and then stops (Shahabudeen *et al.,* 2002). Since production and delivery of each plant is triggered by succeeding plant order, it is called a pull system. In SCS controlled by kanban, the flow of components is from preceding plant to succeeding plant, but information flow is upside down and from succeeding plant to preceding one as shown in Fig. 1.

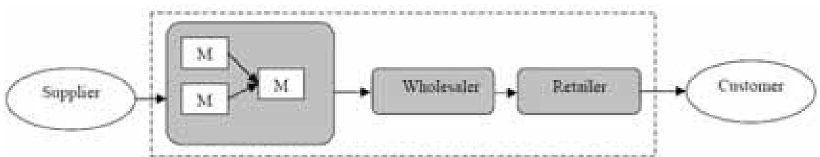

Fig. 1: A typical supply chain system

Just-in-time production system is the product manufacturing philosophy in which right quantity at right time is mostly required (Hutchins, 1993; Shahabudeen *et al.,* 2002). Kanban coupled with pull system of production is used as a means of implementing JIT. To emphasize the importance of kanban number, we quote fukukwa & hong word:

"Since the number of kanbans influences the product inventory level, it is very important for managers to determine the number of kanbans used in the system's introductory phase. Moreover, the number of kanbans can significantly influence the load balance between processes and the amount of orders needed to be obtained from subcontractors."

The rest of this paper is organized as follows: In Section 2, problem definition is presented. While in Section 3, we present an overview of the optimization stages through which modeling specialists go to improve their models. Additionally, GAs are discussed in general to point out their strong features for model optimization and design. Literature review is presented in Section 4. In Section 5, we show one SM example which is used as an illustration. In Section 6, a simulation optimization problem is formulated, and a proposed methodology is presented. Section 7 summarizes the results of the optimization performed. Finally, Section 8 draws conclusion.

Problem Definition:

The function of kanban is explained through the use of an N-stage production system as shown in Fig. 2, where two adjacent plants, i and $i+1$, are isolated for illustrative purpose in Fig. 3. As you see, when plant $i+1$ takes and uses a container, withdrawal kanban is detached and put in kanban post. Then in fixed or unfixed intervals, withdrawal kanbans are collected from kanban post, and accompanied by empty containers to be transshipped to preceding plant. Then all collected withdrawal kanbans and empty containers are put in their particular places. Each detached withdrawal kanban acts as a trigger of preceding plant and commands operators to produce and fill empty containers. At this time, containers filled in store A should wait. Then every filled container transshipped with its kanban, is carried to succeeding plant and put in store B. This cycle repeats.

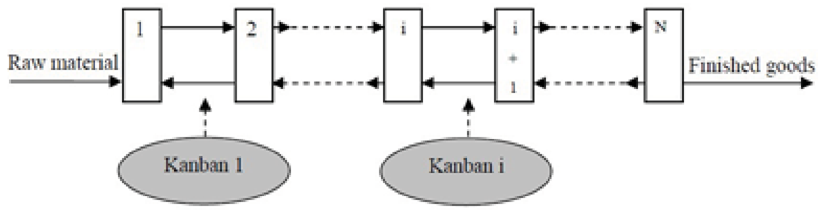

Fig. 2: A multi-stage supply chain system with kanban operation

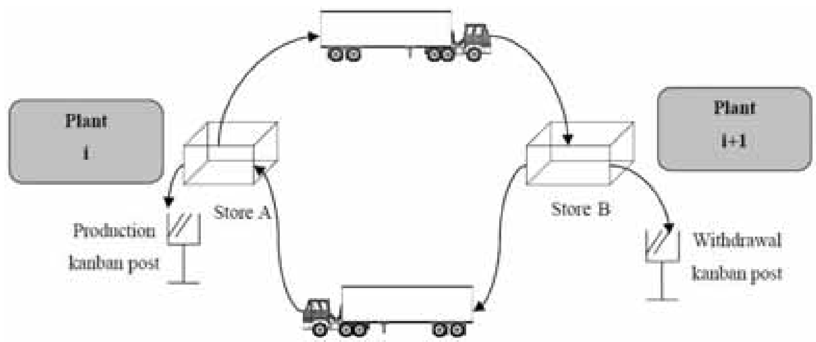

Fig. 3: Operation of kanban production system

Gas and Simulation Optimization:

Model designers build simulation models that can solve so-called forward problems – given the input parameters and the model structure to measure the output or model performance. In most cases, this analytical forward problem is hard enough to solve, so rarely the backward problem is posed. Hence, the desired output is given to find the parameters and/or the model structure which has produced the output. This backward problem is non-analytic and often seems to be even more complex than the forward problem. The backward problem, even when it's not obviously manifested, is often solved by primitive optimization techniques, like Trial and error or Manual Gauss–Seidel method that in both methods, designers are mediator between model and algorithm, and change the parameter manually.

However, Real optimization starts when the designer who searches in the parameter space, is not a mediator between model and algorithm, but when the searching process is done automatically. An optimization algorithm is coupled to the model and changes control parameters of the model directly. The result of the simulation run is returned back to the model, and it decides what new parameter should be changed to increase model performance.

GAs recommend themselves as universal tools for dealing with backward problems. In this research, we apply GA to solve demand problem that is explained as follows.

Literature Review:

Supply chain management (SCM) is defined as a set of approaches applied to integrate suppliers, manufacturers, warehouses and stores efficiently, so that goods are produced and distributed in right quantities, to right locations, and at right time in order to minimize system-wide costs while satisfying service level requirements (Simchi-lei *et al.,* 2000). Since supply chain management has drawn much attention in industrial and academic fields, various techniques are developed to model, analyze and solve complex decision making problems in supply chains. Simulation is one of those techniques, which allows researchers to capture and experiment with the rules in real or proposed systems. Oftentimes, there are some situations in which a problem cannot meet the assumptions set by analytical modeling methods, especially when a problem exhibits significant uncertainty and is quite difficult to be handled analytically (Evans and Olson, 1998). Computer-based simulation with its own strength on evaluating variations and interdependencies in a complex system is one of the promising methods (Wyland, 2000). With simulation, it is possible for decision makers to examine the changes in some parts of the chain and following consequences with less expenses than field experiment which is usually difficult to be carried out. For example, Kimbrough *et al.* (2002). developed a multi-agent simulation system and showed that with proper feedback control design, artificial intelligence could help tiers find an optimized inventory policy to mitigate bullwhip effects. Minegishi and Thiel (Minegishhi and Thiel, 2000) simulated complex logistic behaviors of an integrated food industry and proposed several managerial recommendations from simulation results. On this topic, Chatfield *et al.* (2004) analyzed the influence of lead-time variability on firm's inventory management by a multi-agent simulation system, while Zhang *et al.* (2006) studied how information could reduce the influence of shipment uncertainty during the lead time by a discrete simulation model. Importance of kanban systems was first brought to light by Mondon in 1983, and was the basis for many researches. He presented a summary of Toyota approach to determine the number of kanbans. Brekley 1992 reviewed fifty papers in the field of production control via kanban and categorized them on the base of their systems. He listed 24 design parameters for kanban performance as well. By combining just-in-time and simulation, Azadeh *et al* (2005) introduces a framework to redesign manufacturing systems into practical optimum just-in-time systems through integration of computer simulation and analysis of variance.

Simulation Model:

The supply chain system model presented in this paper includes five stages, i.e., SCS includes supplier, factory, wholesaler, retailer and customer. The simulation model is shown in Fig. 4. A brief explanation is presented as follows:

The production time in each station is denoted by P_i for $i=1,2,3$. The time between arrivals of the demand for final product is denoted by D and each demand includes B_4 products. The reader should note that Pi and D could be either deterministic or randomly distributed. The transportation time between any two neighboring stations is assumed to be zero. The lot size (inventory) for production and transportation of each kanban to each station is denoted by $B_1,...,B_4$, that should be determined by GA. Therefore, each kanban indicates production or transfer of B_i units of inventory in the system. If an order is received and there is no inventory to satisfy the demand, the order should wait. At the beginning of the work in this system, enough units of inventory in some of the QUEUE nodes are considered to prevent instability. The main structure of this model is composed of ASSEMBLY (SELECT), QUEUE, BATCH and GOON nodes. The role of a QUEUE node is to store the final product or a product waiting for process. A GOON node branches the flow of entity. On the other hand, An ASSEMBLY node combines the received kanban with required inventory and then gives the authority for production or transfer of the processed product to the next station. For transferring, the processed products should firstly be batched in their batch sizes ($B_1,...,B_4$). A BATCH node perform this task in VISUAL SLAM. The information flow for production orders and receiving materials are shown as the connections between GOON and QUEUE nodes. The first entity that starts production takes place by receiving the first order at the CREATE node. The order occurs in the CREATE node and is transferred to the UNBATCH node, NBD, to translate demand number to order quantity and also if there is no product in *QIF3*, it sends one massage to be produced. Then, the transferred and received orders in *QOF3* node and the inventory in *QIF3* node are combined in the ASSEMBLY node, *AS4*, and the assembled entity leaves the system. Simultaneously, the GOON node duplicates the entity. One of the duplicated entities inserts to the GOON node, *GPK3*, to authorize the production of another product in the last station by transferring one entity to *QPK3* and also, if there is no material in *QINF3*, it orders semi-finished product to the previous station. The other one inserts to the BATCH node for batching and transferring to customer. The entity at the *QPK3* node is assembled with one unit of inventory in the *QINF3* node by the ASSEMBLY node, *AP3*. The assembled entity is then duplicated by GOON node. One of the duplicated entities will wait in the AWAIT node, *AOP3*, to be processed by operation P_3 and the other one will be batched in B_3 size and then authorized to transfer B_3 units of semi-finished product from the previous station to the next. This is accomplished by the entity waiting in the *QWK3* node to receive a semi-finished product and assembling it with one unit of inventory in the *QIF2* node through the ASSEMBLY node *AS3*. Then, batching semi-finished product and when the number of products in the batch achieves to the batch size (B_3) are transferred to the main process. After operation P_3 on the waiting entity in the AWAIT node, *AOP3*, the FREE node will release the resource named OP_3. The process will be continued up to the beginning of the network representing the arrival of raw materials to the system (Azadeh *et al.,* 2005).

Control Variables:

The Simulation Model has many parameters that could be used as control variables for optimization. Considering the simulation model and SCS goals, we decided to choose $B_1,...,B_4$ to control our variables. Variation in every $B_1,...,B_4$, causes variation in output statistics of VISUAL SLAM.

Output Criteria:

Many statistics could be gathered through running this model under different scenarios. To keep the interpretation simple, we chose only two measures representing scenario's quality as hereunder:

1- Average WIP in every queue during the simulation.
2- Sum of transshipments during the simulation.

However, the main objective of just-in-time is one product flow. But in SCS, when two neighboring stations are distant, specialty causes high transshipment cost. In fact, inventory holding cost of WIP in SCS has inverse relation with transshipment cost. Consequently, if GA goal function is the function of both above parameters, it will have at least one optimum point.

Goal Function = f (Inventory holding cost, Transshipment cost)

Actually if we want to clear our purpose of simulation optimization via GA, we can state that our purpose is to minimize the GA goal function by considering $B_1,...,B_4$ parameters.

Simulation Horizon:

Every goal function evaluation requires simulation model of SCS run, so it is important to keep the running time as short as possible. On the other hand, the model has its stochastic behavior, hence a long enough running time is needed to make the measured statistics reliable.

We measured the model performance for a time horizon of 5 days, i.e., 60 min×24 h×5 days (the time step of the simulation model is 1 minute). This time horizon is acceptable for the computer we used and the optimization dynamic could be easily observed. Furthermore, we can clear statistics of earlier time of simulation run by using statement MONTOR and choosing the *clear* option up to 10 hours at the beginning of simulation run. This causes to achieve stability of the model faster.

Proposed Methodology:

The genetic algorithm (GA) proposed by Holland (1975) was a meta-heuristic (a heuristic that uses another heuristic) applied to solve various problems including combinatorial optimization (Onwubolu and Mutingi, 2001). GA makes the initial population evolve toward a population that is expected to contain the best solution. It uses the following reproduction–evaluation cycle for each iteration referred to as generation. Chromosomes (individuals) from the current population are selected with a given probability and the copies of these chromosomes are created. The selection process of chromosomes is based on their fitness for current population, i.e., chromosomes with higher quality (less fitness value) will have higher probability of being copied (Pierreval and Tautou, 1997). The fitness is a function of simulation model's response. Selected chromosomes are subjected to mutation and crossover. The process which contains selection, mutation, and crossover is called reproduction. The crossover mechanism allows to mixing parental information to pass through their descendants (offspring). Mutation introduces innovation to the population. From one generation to the next, the population globally tends to have a better fitness (Paris and Pierreval, 2001). The proposed GA-simulation approach is a random search technique. It follows the efficient genetic operations and uses discrete-event simulation to evaluate the fitness (objective value) of each chromosome toward the optimization of the problem. Although GA can solve an optimization problem effectively, it is often sensitive to its search parameters. Its key search parameters are population size, stopping criterion, crossover rate, and mutation rate. The proposed methodology is shown in Fig. 5.

Initial Population:

The first step is to generate initial solutions as an initial population. In order to do this, a number of *k* initial solutions are generated. For the underway problem, each chromosome (a candidate solution) includes 4 genes. The gene at the first position of the chromosome represents batch size for the first stage of JIT simulation model. The gene at the second position represents the batch size for the second stage of JIT simulation model, and so on.

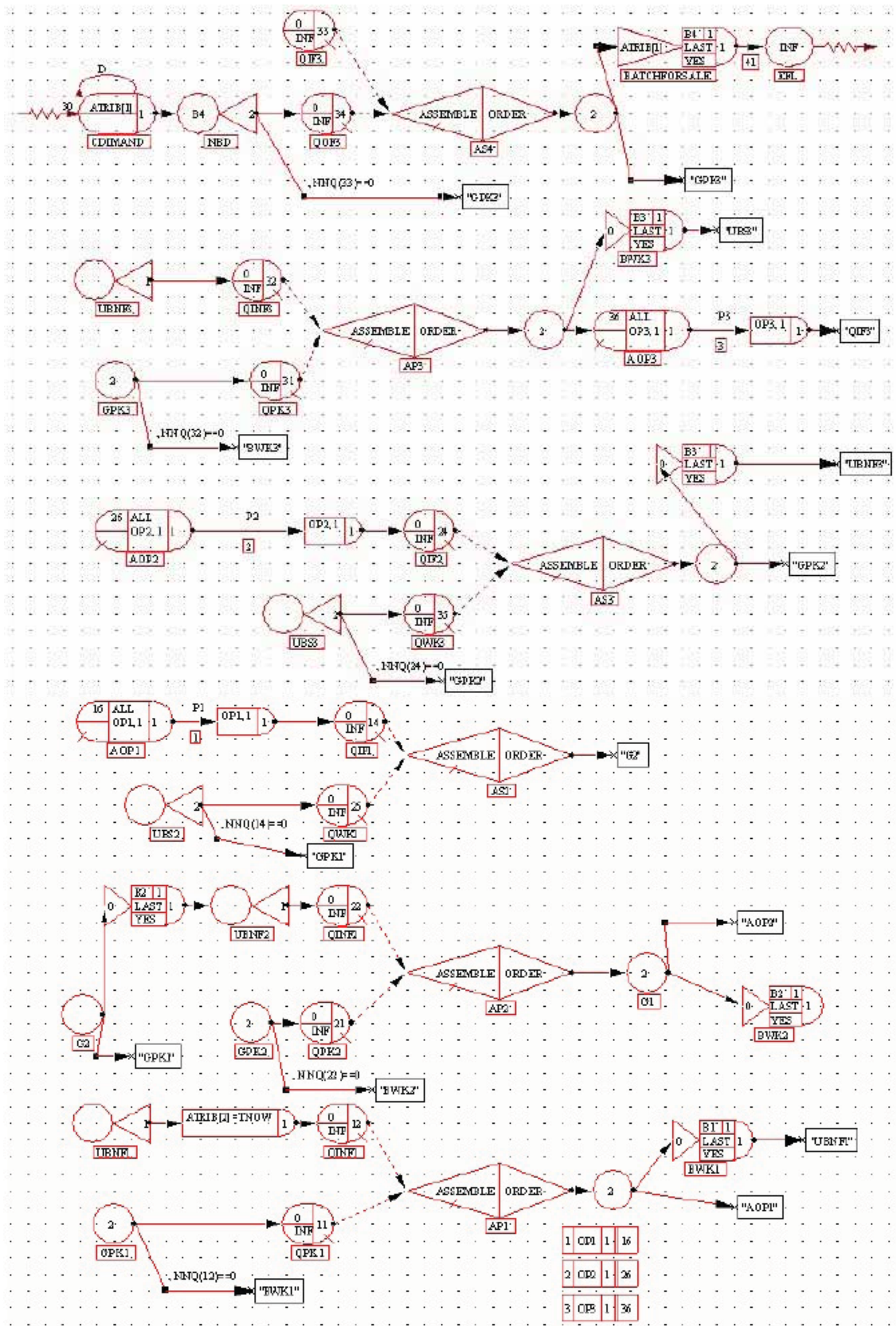

Fig. 4: VISUAL SLAM NETWORK

Performance Evaluation:

A discrete-event simulator was used to evaluate the performance or fitness of each chromosome. As mentioned before our GF is: *Goal Function = f (Inventory holding cost, Transshipment cost).* Here is the definition of two parameters for calculating inventory holding cost and transshipment cost:

H: holding cost of work-in-process inventory in dollar/(unit.year);
A: shipping cost in dollar/ship

To compute inventory holding cost of WIP, it's enough to have the sum of each WIP's total time and the number of all transshipments during the simulation:

$$GF = H * \left(\sum Total\ time\ in\ system\ for\ each\ semi-finish\ good\right) + A(total\ transshipment)$$

Moreover, it is possible to compute our GF in simulation network by ASSIGN node and COLOCT node. It is obvious that total time in system and total transshipment have inverse relation. Therefore the GF mentioned above has at least one optimum point.

Selection:

The selected population in the previous generation is the best 10 individuals of the current generation; it is the elitist strategy of our algorithm. Selecting parents for crossover and mutation is described below:

Crossover:

In this step, we attempt to improve some worse solutions (solutions with the highest fitness function value). Therefore we measure a convex combination of two worse solutions. Furthermore, we do it with the best and the worst solutions and evaluate the value of the resulted offspring's fitness function, i.e., we first sort all the existent solutions and choose the two last solutions as worst solutions and again the last and the first solutions as the best and the worst ones. For more illustration, suppose we have chosen two solutions, Chromosome 1 and Chromosome 2, and moreover λ-value in convex combination is considered to be 0.5. So the new solution can be obtained as follow (Fig. 6).

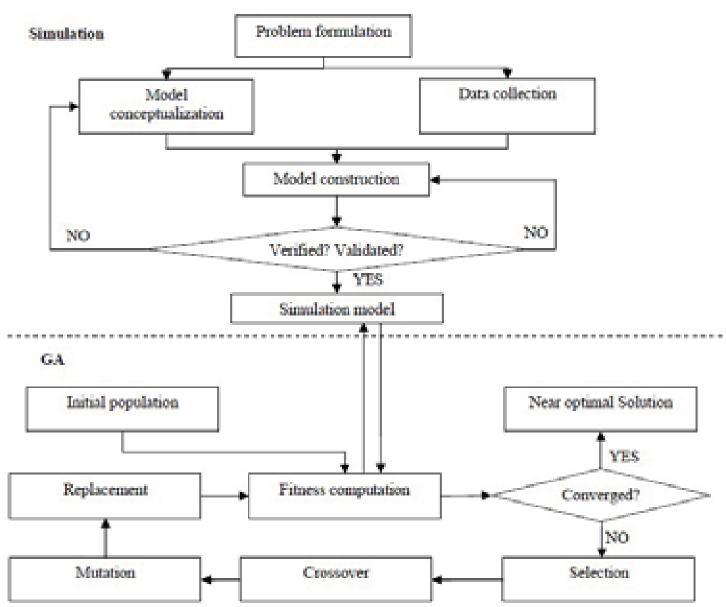

Fig. 5: Relationship between simulation model and GA.

Mutation:

The best and the worst chromosomes generated so far are selected to undergo mutation procedure. For each gene of the chromosomes, first we calculate the average of batch sizes (B_1,...,B_4) and then, select the gene with maximum variation from the average. Afterward, we replace the average of batch sizes with that gene.

That occurs because in accordance with SCS which is controlled by JIT, production line is approximately balanced; So we shouldn't have high variation in B_1,..,B_4. This variation causes high inventory holding cost.

Stopping Criterion:

The evaluation of one chromosome required one simulation run. Thus, the total number of required simulation runs is the multiple of the population size and the number of GA search generations (Taho *et al.,* 2007).

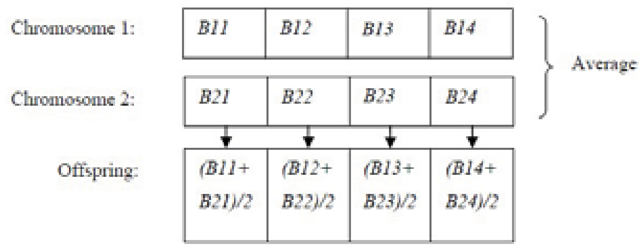

Fig. 6: Crossover

We use a very simple criterion: when a predefined maximum number of function evaluations is reached, or 2- in 20 continuous generation, there is no improvement in best solution.

Simulation Results:

For evaluation if model and mention method we use V.S software V.B programming. Finally the best solution (optimum solution) obtained from simulation model via GA was achieved as follows:

$B_1=9,\ B_2=8,\ B_3=7,\ B_4=8;$

As mentioned before, because production line is approximately balanced, we expect to have low variation in batch sizes B_1,..,B_4. Since we have no prior knowledge of SM behavior, this low variation itself illustrates the quality of optimization solution.

Conclusion:

This paper has presented a GA-simulation approach to solve batch size problem for kanban in a supply chain system under just-in-time philosophy. It has combined stochastically modeling capability of discrete-event simulation and the intelligent search algorithm of GA for supply chain system which is controlled by just-in-time philosophy. Because of stochastic behavior of SCS parameters, one of the best way to model such problems is simulation. On the other hand, unlike many other optimization methods, GA is successfully applied to solve optimization and design problems, though the model has stochastic nature. Among all optimization methods, GA provides the maximum ''black-box'' approach, i.e., no preliminary consideration about the goal function or initial value of the control parameters needs to be taken. This is important in Simulation Model area where no prior knowledge of SM behavior may exist like this problem. It is claimed that GA can be used as a standard optimization technique with any Simulation Model.

REFERENCES

Andradottir, S., 1992. Discrete optimization in simulation: A method and applications, in: J.J. Swain, D. Gainsman (Eds.), Proceedings of the 1992 Winter Simulation Conference, SCS, San Diego, CA., pp: 483–486.

Azadeh, A., B. Bidokhti, S.M.R. Sakkaki, 2005. Design of practical optimum JIT systems by integration of computer simulation and analysis of variance, Computers & Industrial Engineering, 49: 504–519.

Chatfield, D.C., J.G. Kim, T.P. Harrison, J.C. Hayya, 2004. The Bullwhip effect – impact of stochastic lead times, information quality, and information sharing: a simulation study, Production and Operations Management, 13(4): 340–353.

Evans, G.W., B. Stuckman, M. Mollaghasemi, 1991. Multicriteria optimization of simulation models, in:B.L. Nelson, W.D. Kelton, G.M. Clark (Eds.), Proceedings of the 1991 Winter Simulation Conference, SCS, San Diego, CA., pp: 894–899.

Evans, J.R., D.L. Olson, 1998. Introduction to Simulation Chapter 1, Introduction to Simulation and Risk Analysis, Prentice-Hall, Upper Saddle River, N.J., pp: 1–18.

Kimbrough, S.O., D.J. Wu, F. Zhong, 2002. Computers play the Beer Game: can artificial agents manage supply chains? Decision Support Systems, 33: 323–333.

Minegishi, S., D. Thiel, 2000. System dynamics modeling and simulation of a particular food supply chain, Simulation Practice and Theory, 8: 321–339.

Onwubolu, G.C., M. Mutingi, 2001. Optimizing the multiple constrained resources product mix problem using genetic algorithms. International Journal of Production Research, 39(9): 1897–1910.

Pierreval, H., L. Tautou, 1997. Using evolutionary algorithms and simulation for the optimization of manufacturing systems. IIE Transactions, 29(3): 181–189.

Paris, J.L., H. Pierreval, 2001. A distributed evolutionary simulation optimization approach for the configuration of multiproduct kanban systems. International Journal of Computer Integrated Manufacturing, 14(5): 421–430.

p.shahabudeen, R. Gopinath, K.krishnaiah, 2002. Desighn of bi-criteria kanban system using simulated annealing technique.

Shajun Wang, Bhaba Sarker, 2004. Optimal model for supply chain system controlled by kanban under just-in-time philosophy.

Stuckman, B., G. Evans, M. Mollaghasemi, 1991. Comparison of global search methods for design optimization using simulation, in: B.L. Nelson, W.D. Kelton, G.M. Clark (Eds.), Proceedings of the 1991 Winter Simulation Conference, SCS, San Diego, CA., pp: 937–943.

Tomkins, G., F. Azadivar, 1995. Genetic algorithms in optimizing simulated systems, in: C. Alexopoulos, K. Kang, W.R. Lilegdon, D. Goldsman (Eds.), Proceedings of the 1995 Winter Simulation Conference, SCS, San Diego, CA., pp: 757–762.

Taho Yang, a., Yiyo b. Kuo, Chiwoon Cho, 2007. A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem European Journal of Operational Research, 176: 1859–1873.

Simchi-levi, D., P. Kaminsky, E. Simchi-levi, 2000. Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies,Irwin/McGraw-Hill, Boston.

Wyland, B., 2000. Simulating the supply chain, IIE Solutions, 32(1): 37–42.

Zhang, C., G.W. Tan, D.J. Robb, X. Zheng, 2006. Sharing shipping quantity information in the supply chain, Omega: International Journal of Management Science, 34(5): 427–438.