Optimisation of multi-stage supply chain systems by integrated simulation-variable neighbourhood search algorithm

Ali Azadeh* and Ali Zahedi Anaraki

School of Industrial Engineering and
Center of Excellence for Intelligent Based Experimental Mechanics,
College of Engineering,
University of Tehran, Iran
Fax: +98-21-82084194
Email: aazadeh@ut.ac.ir
Email: ali@azadeh.com
Email: amir.zahedy@gmail.com
*Corresponding author

Sara Motevali Haghighi

Department of Engineering,
Esfarayen University,
North Khorasan, Iran
Email: sara.motevali@ut.ac.ir

Zahra Jiryaei

Department of Industrial Engineering,
College of Engineering,
University of Yazd, Iran
Email: z.jiryaei@yahoo.com

Fatemeh Nadarlou

Department of Industrial Engineering,
College of Engineering,
University of Tafresh, Iran
Email: f.nadarlou63@gmail.com

Abstract: In this paper, multi-stage supply chain systems (SCSs) controlled by kanban system are appraised a new simulation metaheuristics approach. In the kanban system, decision making is based on determination of batch size for each kanban. This paper simulates 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. This paper applies genetic algorithm (GA) and variable neighbourhood search (VNS) to optimise the simulation model. A simple real-coded GA and VNS 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 and VNS and GA-VNS successfully provide a parameter set to demonstrate its capability to solve such difficult backward problems even in the area of complex simulation model optimisation specially when there is no prior knowledge of simulation model behaviour.

Keywords: supply chain system; SCS; just-in-time; JIT; computer simulation; optimisation; genetic algorithm; GA; variable neighbourhood search; VNS; services management; operations management.


Biographical notes: Ali Azadeh is an Eminent University Professor and Founder of Department of Industrial Engineering and co-founder of Research Institute of Energy Management and Planning at University of Tehran. He obtained his PhD in Industrial and Systems Engineering from the University of Southern California. He received the 1992 Phi Beta Kappa Alumni Award for excellence in research and innovation of doctoral dissertation in the USA. He is the recipient of six awards at the University of Tehran. He is also the recipient of National Eminent Researcher Award in Iran. He has published more than 650 papers in reputable academic journals and conference proceedings.

Ali Zahedi Anaraki received his MS in Industrial Engineering from School of Industrial Engineering, University of Tehran, Iran. His research interests span from scheduling, data mining, multi-attribute decision making to combinatorial optimisation.

Sara Motevali Haghighi received her MS in Industrial Engineering at the School of Industrial Engineering, University of Tehran, Iran and her BS in Industrial Engineering at Bojnord University, Iran. Her research interests span from scheduling, data mining, multi-attribute decision making problem and simulation problems to combinatorial optimisation.

Zahra Jiryaei is a PhD candidate at Yazd University, Iran. She received her MS and BS degrees from Tafresh University, Iran.

Fatemeh Nadarlou received her MSc in Industrial Engineering from University of Tafresh, Iran. Her research interests span multi-attribute decision making to simulation.

1 Motivation and significance

An important issue in industrial and academic field is supply chain management (SCM). SCM focuses on planning and forecasting, purchasing, product assembly, moving, storage, distribution, sales and customer service. SCM professionals are involved in every facet of the business process as they strive to achieve a sustainable competitive advantage
Optimisation of multi-stage supply chain systems

by building and delivering products better, faster and cheaper. Organisations focus on increasing productivity, improving the quality of its products, and raising the standards of efficiency within its company to remain competitive and economically successful. The just-in-time (JIT) management allows the organisation to achieve this goal by increasing the efficiency of the production, reducing the level of wasted materials, time, and effort involved in the production process. To implement the JIT philosophy, a kanban technique is introduced as an efficient operational mechanism. There are several methods for modelling, analysing and solving complex decision making problems in supply chains. Computer simulation is widely used for solving complex supply chain problem. Some parameters of supply chain system (SCS) are stochastic. So, simulation is the best technique for modelling of this problem. Also, because of stochastic nature, hybrid genetic algorithm (GA)-variable neighbourhood search (VNS) is applied to solve optimisation and design problems. This paper is presented a new approach based on computer simulation, just in time, GA and VNS for optimisation of these systems. JIT philosophy and kanbans are used for controlling of system.

2 Introduction and literature review

Simulation is one of the widely used modelling tools to analyse different systems. But, this technique does not alone for analysing the system. Simulation models (SMs) can be optimised difficult or even impossible problem. In this paper, the successful application of GA was demonstrated to SM optimisation and design (Azadeh et al., 2010a; Andradottir, 1992; Evans et al., 1991; Stuckman et al., 1991; Tomkins and Azadivar, 1995). Berkley (1991) developed SMs to determine the minimum number of kanbans to find a desired production rate for tandem queues. The models were further modified in to a two-card kanban controlled system (Berkley, 1996). He concluded that the optimal number of kanbans can be determined as trade-offs between costs of frequent material handling and the benefits of lower work-in-process (WIP) levels. Karmarker and Kekre (1989) and Yavuz and Satir (1995) demonstrated that reducing the kanban size improves the performance of the system by lowering the inventory levels and the make span, or in other words, increasing the kanban size will accumulate WIP, but improves the fill rate, which provides better customer service. Savsar (1997) conducted a study on the optimal kanban size for production in an electronic assembly line. A cost model in to a two card kanban system with a fixed kanban size was introduced by Ohno et al. (1995) to determine the optimal number of kanbans. The objective of this algorithm was to minimise the expected cost per period. It was shown in the results that a certain value of the number of kanbans can achieve the lowest average cost by balancing the backlogged inventory and ordering cost. Gupta and Al-Turki (1997) introduced a systematic methodology to manipulate the number of kanbans in a JIT system, where an algorithm to minimise the backlog and WIP was developed for stochastic processing times and variable demand environment. Miyazaki et al. (1988) derived two formulae to calculate the average inventory yielded by fixed-interval withdrawal kanbans and supplier kanbans in a JIT production system, and the minimum number of kanbans required for this system was determined by two formulae. Sarker and Balan (1999) determined the number of kanbans that required to transport materials between two workstations for both single-stage and multi-stage kanban systems. In their models, the demand rate was assumed as linear over each of the three phases (inception, maturation and declination) of
A. Azadeh et al.

a product’s life cycle. This model eventually determined batch sizes (or the container size), number of kanbans, the dispatching time intervals, and the schedule for production. Mascolo et al. (1996) developed a general-purpose analytical method for performance evaluation of multi-stage kanban controlled production systems. Nori and Sarker (1998) determined the delivery policy and the number of kanbans between two workstations. In their models, the total cost was expressed as a function of the number of kanbans, shortage cost of materials, and holding cost of containers. Tardif and Maaseidvaag (2001) proposed an adaptive kanban system. This system monitored the actual inventory level for adjusting the number of kanbans. Takahashi and Nakamura (1999, 2000a, 2000b, 2002) proposed reactive JIT ordering systems to distinguish stable changes from unstable changes. Takahashi et al. (2004) proposed a reactive JIT ordering system for multi-stage production systems with unstable changes in demand, not only in the mean but also in the variance. Chan (2001) used a computer SM to investigate the effect of varying kanban size on the performance of JIT manufacturing systems. Erhun et al. (2003) proposed an analytical model to simultaneously determine the withdrawal cycle length and the size of kanbans in a multi-item, multi-stage, multi-period, capacitated and periodic review kanban system. Moattar Husseini et al. (2006) proposed a method which uses an integer linear programming technique to flexibly determine the number of kanbans for each stage of a JIT production system, by minimising total inventory cost for a given planning horizon. Wang and Sarker (2006) proposed a branch and bound method for a multi-stage SCS to define the optimal number of kanbans, the batch size, the number of batches and the total quantity over one period. Yang et al. (2007) addressed an evolutionary-simulation optimisation approach in solving a multi-constant work-in-process (multi-CONWIP) pull strategy problem. In general, the proposed methodology was effective and robust for the proposed problem. Qi and Weiming (2008) proposed a hybrid simulation approach, using both discrete event and agent-based technologies to model complex material handling processes in an assembly line. The proposed hybrid modelling approach facilitates the implementation of a responsive and adaptive environment in that various ‘what-if’ scenarios can be simulated under different simulation configurations and real-time situations. Azadeh et al. (2005) proposed a framework based on combining JIT and simulation to redesign manufacturing systems to achieve practical optimum JIT systems by integration of computer simulation and analysis of variance.

Berkley (1992) reviewed 50 papers in the field of production control via kanban and categorised them on the base of their systems. He listed 24 design parameters for kanban performance as well. Azadeh et al. (2010a) evaluated multi-stage SCS controlled by kanban system. In their study, decision making is based on determination of batch size for each kanban. They simulated SCS regarding the costs under JIT production philosophy. They applied simulation and GA and simulation together for improving kanban batch size. Azadeh et al. (2010b) developed a hybrid approach involving GAs as an optimisation search technique and a SM representing the dynamic behaviour of a system and its limitations to improve an existing JIT manufacturing system. To achieve the objective, the existing system is modelled and simulated (by considering the system’s limitations and its dynamic behaviour). Then, the integrated SM is tested and validated by analysis of variance.
In today’s competitive and dynamic business environment, companies are forced to find all the ways to enhance the product availability, increase the product variety and reduce inventory to achieve better performance in the supply chain. One may go for one way product substitution in the supply chain for increasing product availability and supply chain profit. Routroy et al. (2011) considered a SM both, with product substitution and without product substitution is developed in ARENA in order to study the impact of one way product substitution on supply chain performance. Song et al. (2010) applies the methodology of simulation meta model to a multi-echelon supply chain problem and conduct statistical analysis of the parameters.

Haughton and Sapna Isotupa (2013) examined the traffic control tactic of reducing the amplitude of hour-to-hour arrival rate of commercial traffic at border checkpoints between Canada and the USA. They used a simulation study to show that this tactic can facilitate attainment of two important government priorities: minimise unnecessary investment to expand border infrastructure and minimise the risk of trans-border.

Green et al. (2014) studied about the impact of a total JIT strategy in supply chain environment. Their results showed the significant relationships between a SCM strategy and total JIT. Zamarripa et al. (2012) applied a GA for solving supply chain. They considered uncertainty environment for supply chain problem. Costa et al. (2010) proposed a new encoding-decoding procedure for GA algorithm to minimise total cost. Cigolini et al. (2014) combined configuration and performance for supply chain problem. They applied simulation and statistical analysis for their problem. Shi et al. (2013) presented discrete-event simulation for supply chain problem. JIT and uncertainty logic was considered by them. Kuo and Han (2011) considered bi-level linear programming. Also, they applied hybrid GA and particle swarm optimisation (PSO). Srinivas et al. (2008) considered single-vendor-multi-buyer consignment stock policy for supply chain problem. They used GA to minimise total expected cost of vendor and buyer. Yan et al. (2012) proposed a demand model with Poisson process. Their model can be applied for supply chain models. Zhang et al. (2013) presented a multi criterion problem for order distribution of supply chain. They applied GA for optimisation. Finally, a numerical example showed the efficiency of their proposed model. A multi-agent SM was proposed by Li et al. (2010) for analysing of supply chain problem. They simulated raw material suppliers and component supplier. Asar et al. (2006) considered risk for supply chain problem and applied fuzzy logic. Sobhani and Wong (2013) optimised distribution quantity of products in a three-stage SCS. They applied mathematical model and GA for minimisation cost. Phonsuwan and Kachitvichyanukul (2013) studied one-Tambon-one-product supply chain. They applied discrete event simulation for their system with hybrid push-pull flow control. Parveen and Rao (2009) proposed an integrated approach for designing and analysing of lean supply chain with JIT production. Grewal et al. (2010) compared the reorder point and kanban replenishment strategies and their performance is evaluated under the optimal decision variable settings for each strategy. Today, enterprises are under pressure to improve process performance while still remaining customer oriented. The problem of workflow and process quality is a very important issue in SCM. Successful improvement of the logistics process has to be viewed as one way of making improvements to the integrated supply chain network. Ho
et al. (2010) described the application of an online analytical processing (OLAP)-based neural ensembles strategy to the acquisition of process knowledge during the supply chain operations. Zegordi et al. (2010) considered the scheduling of products and vehicles in a two-stage supply chain environment. $m$ suppliers with different production speeds were in the first stage, while there were $n$ vehicles with different speed and different transport capacity in the second stage. They modelled the situation as a mixed-integer programming (MIP) problem. Then, problem was solved by GA algorithm. Supply chain demand is often prone to fluctuations and instability. Known as the ‘bullwhip effect’, small variations in end-item demand create order and inventory oscillations that amplify from a downstream site to an upstream site. Applying a risk analysis approach, and assuming the bullwhip phenomenon as a constant reality, Thierry et al. (2011) presented the profits or losses that can accrue from various cooperation policies. The system considered for this research is a four-stage supply chain. To allow risk measures and analysis, a specific discrete-event-simulation system was developed.

Mizgier et al. (2012) introduced an agent-based model of a supply chain network which represented in more detail the real economic environment in which firms operate. They focused on the influence of local processes on the global economic behaviour of the system and study how the proposed modifications change the general properties of the model. In Khalili-Damghani and Taghavifard (2012), a generic process in which JIT practices are changed into agility indices, and agility indices are converted into performance measurement in supply chain has been supposed in form of a conceptual model. In order to embrace the lean principles, supply chains of service companies (SCSCs) likely have to modify how they manage operations. Portioli-Staudacher and Tantardini (2012) investigated which could be the main problems that managers of SCSCs may encounter in modifying how they manage operations to be more in line with the lean approach next to presenting a new framework for the defining characteristics of lean in service companies.

Aldaihani and Darwish (2013) considered a supply chain with a single manufacturer supplying a newsvendor-type item to multiple retailers (newsvendors). In Mousavi and Tavakkoli-Moghaddam (2013) presented a two-stage MIP model for the location of cross-docking centres and vehicle routing scheduling problems with cross-docking due to potential applications in the distribution networks. Then, a new algorithm based on a two-stage hybrid simulated annealing (HSA) with a tabu list taken from tabu search (TS) is proposed to solve the presented model. Optimisation problems in SCM are commonly cast as large scale mixed-integer linear programs (MILPs) that are hard to solve in short CPU times. Copado-Mendez et al. (2013) presented a novel solution method for this type of problems that combined the strengths of standard branch and cut techniques with the efficiency of large neighbourhood search (LNS). Shankar et al. (2013) aimed at multi-objective optimisation of single-product for four-echelon supply chain architecture consisting of suppliers, production plants, distribution centres (DCs) and customer zones (CZs). In Duan and Liao (2013), the optimal replenishment policies of capacitated supply chains (SC) operating under two different control strategies (decentralised vs. centralised) and various demands are determined and insights useful to management are discussed.
Figure 1  The framework of this paper

Section 1:  
Introduction  
Literature review

Section 2:  
Problem description

Section 3:  
Methodology

Section 3.1:  
Description of genetic algorithm and variable neighbourhood search

Section 3.2:  
Chromosome encoding

Section 3.3:  
Description of simulation model

Section 4:  
Numerical example and analysis of solution

Section 5:  
Conclusions
In this field with studying literature review it is obvious that VNS algorithm is not yet applied along with simulation. Since VNS algorithm is a new and efficient algorithm in obtaining almost exact solution it is very useful for improving solutions in just in time system. The goal of this study is to apply GA as an optimisation stochastic search technique with the capability to deal with a large solution space (e.g., the solution space of the feasible number of kanbans in each station) to simulate JIT model for determination the optimal/near-optimal number of kanbans in each station of the practical JIT system. Since design and implementation of theoretical JIT philosophy may not be possible for most dynamic systems; due to their unique limitations and constrains, a more applicable JIT design approach which is compatible with the limitations of the dynamic system is called practical JIT (Azadeh et al., 2005). Figure 1 depicts the framework of this research. The rest of this paper is organised as follows: Section 2 provides description of problem. The approach including GA and simulation is presented in Section 3. Section 4 provides numerical example and analysis of solution. Finally, the conclusions are presented in Section 5.

3 Problem description

Kanban system is applied for N-stage production system when there are two adjacent plants. Withdrawal kanban is disconnected and put in kanban post when one plant takes and uses a container. These withdrawal kanbans are gathered from kanban post in fixed or unfixed intervals. Then empty containers accompanied these cards and transshipped to preceding plant. Finally, withdrawal kanbans and empty containers are placed in their particular places. For kanban system, this cycle is repeated. Withdrawal kanban and production kanban are two types of kanban cards. Figures 2 and 3 show a multi-stage SCS with kanban operation and operation of kanban production system (Azadeh et al., 2010a).
An important issue in industrial and academic field is SCM. There are several methods for modelling, analysing and solving complex decision making problems in supply chains. There are a set of suppliers, manufacturers, warehouses and stores in SCM. Figure 4 show a typical supply chain system (Azadeh et al., 2010a).

The goal of SCN is to minimise system-wide costs by considering satisfied service level requirements (Simchi-levi et al., 2000; Shajun and Bhaba, 2006). High quality production, low cost and right time delivery are important indicators that SCS guarantees these. Appropriate and on time delivery in these systems means that the products must be delivered exactly according to order. There are two types of SCS, single-stage supply chain system (SSSCS) and multi-stage supply chain system (MSSCS). SSSCS consists of two plants but MSSCS consists of more than two plants.

4 Methodology

In this section, proposed approach is presented. GA is applied as an optimisation stochastic search technique with the capability to deal with a large solution space to simulate JIT model for determination the optimal/near-optimal number of kanbans in each station of the practical JIT system.

4.1 GA versus VNS

One of the best ways to solve a problem is genetic algorithms. Also, this model will be able to create a high quality solution. Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem. Genetic algorithms tend to progress in an environment in which there is a very large set of solutions. This algorithm can be applied in any environment.

VNS was proposed as meta heuristic model just a few years ago (is based upon a simple principle: systematic change of neighbourhood within the search). Its development has been rapid, with several dozen papers already published or to appear. Many extensions have been made, mainly to allow solution of large problem instances. In most of them, an effort has been made to keep the simplicity of the basic scheme.

Appendix 1, Appendix 2 and Appendix 3 show procedure GA, VNS and GA-VNS algorithms, respectively. Also, SMs of GA, VNS and GA-VNS are presented in Appendix 4.
4.2 Chromosome encoding

For the problem under consideration, $c$ of four genes (the number of kanbans in each station). Each gene is composed of three bits to determine one of the eight alternatives each gene could obtain (number of kanbans at each station could alter from 1 to 8, e.g., 000 stands for 1, 001 stands for 2 and so on). The gene at the first position of the chromosome represents the number of kanban cards of the first station. The gene at the second position of the chromosome represents the number of kanban cards of the second station, and so on (Table 1).

<table>
<thead>
<tr>
<th>Number of kanban posts ($K$)</th>
<th>$K = 1$</th>
<th>$K = 2$</th>
<th>...</th>
<th>$K = 9$</th>
<th>$K = 10$</th>
<th>...</th>
<th>$K = m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>The solution space</td>
<td>$n^1$</td>
<td>$n^2$</td>
<td>...</td>
<td>$n^9$</td>
<td>$n^{10}$</td>
<td>...</td>
<td>$n^m$</td>
</tr>
</tbody>
</table>

Since the increase in the number of kanbans improves the total cycle time, resource utility and production rate but deteriorate the average inventory in process and the average queue length, the fitness function is defined as equation (1). The weights of the fitness function could be adjusted according to the policy of the factory and the experience of decision makers. In this case, the weights are defined based on the production objectives of the factory, constraints in lay out, machines and work force limitations. The weights are normalised weighted average of the managers’ opinion about the priority of these criteria, given by equation (1)

- **selection**: $p_s = 0.5$
- **mutation rate**: $p_m = 0.15$ (performs mating using single point crossover)
- **population size**: $pop\ size = 16$
- **maximum iteration**: 100
- **fitness function**:

$$0.035(t_i - t_e) + 45(e_i - e_e) - 0.2(l_i - l_e)$$  

where $t_i$ and $t_e$ represent total cycle time of the $i$th chromosome and total cycle time of the practical JIT system, respectively; $e_i$, $e_e$ represent the average resource utilisation in the $i$th chromosome and in the practical JIT system, respectively; and, $l_i$, $l_e$ represent average in-process inventory, (not average queue length, because there is more than one queue in the system), in the $i$th chromosome and in the practical JIT system, respectively. As mentioned it was the managers’ opinion to break down these weights as discussed (0.035, 45 and 0.2 for total cycle time, the average resource utilisation, and average in-process inventory, respectively). However, to illustrate the application of this approach for other studies in actual management systems, ten distinct weights have been chosen. Furthermore, the other users may use the approach of this study according to importance of weights in their production settings.

Once these parameters are determined, the GA is then integrated with the SM to determine the optimal/near-optimal number of kanban cards in each station (i.e., the best chromosome).
4.3 Description of SM

In this paper, five stages were considered for SCS model. This system includes supplier, factory, wholesaler, retailer and customer. Figure 5 shows the SM. A brief explanation is presented as follows:

Figure 5 Simulation model

$P_i$ denote the production time in each station for $i = 1, 2, 3$. Also, the time between arrivals of the demand for final product is denoted by $\text{expon}(10)$ and each demand includes num products. $P_i$ and $D$ could be deterministic or randomly distributed. The transportation time between any two neighbours in stations is equalled to zero. Size of inventory is denoted by $B_1,...,B_4$ for production and transportation of each kanban to each station. This parameter should be determined by GA and VNS. Production or transfer of $B_i$ units of inventory in the system was dictated by each kanban. The order should wait when there is no inventory to satisfy the demand. The main structure of this model is composed of ASSEMBLY (SELECT), QUEUE, BATCH and GOON nodes (Azadeh
et al., 2010a). Table 2 described visual SLAM original network definition of problem. At first, for prevent instability enough units of inventory are considered in some of the QUEUE nodes. Final product or products waiting for process are stored by this node. A GOON node is applied to branch the flow of entity. Then, the received kanban card should be checked with required inventory after that system could start production or of the processed product to the next station that ASSEMBLY nodes is applied. The processed products should be batched in their \((B_1...B_4)\) for simulation BATCH node is used. Connections between GOON and QUEUE nodes illustrate information for material receiving and ordering production. The first entity should be produced for the first order at the CREATE node. Then, this order transfer to the UNBATCH. When there is no products in Q1, this node send message for production. In ASSEMBLY node, AS4 transferred and received orders in QUQU1 node should be combined with the inventory in Q1 node. Then, this entity leaves system. GOON node duplicates the entity that one of them inserts to the GOON node, G1. This node permits to produce another product in the last station by transferring one entity to QUQU2. Semi-finished products to the previous station are ordered, if Q2 was being empty. Batching and transferring to customer for another entity must be done by BATCH node. The entity at the QUQU2 node with one unit of inventory in the Q3 node is combined by the ASSEMBLY node, AP3. GOON node duplicates the assembled entity that one of them inserts to the GOON node. In the AWAIT node, one of the duplicated entities must be wait. Operation T2 starts process for entity in AWAIT1. Other duplicated entities for transferring B3 units of semi-finished product will be batched. The entity in the QUQU4 node gives a message for receives a semi-finished product. Then inventory in the QUQU3 node must be assembled by ASSEMBLY node AS3 transferred to the main process. FREE node will release the resource. This process will be continued up to the beginning of the network (Azadeh et al., 2010a).

5 A numerical example

The GA and VNS and GA-VNS SM is implemented by c# software. At the beginning of program, there are the following forms that are shown in Figures 6 and 7. At first, this form is defined according to the following definitions and delineations:

1. The numbers of program step which can be consisting of manufactory and job shop and distributor and retailer. It is considered in this problem equal with 4.
2. The order rate for customers is considered to be expon (10) in this problem.
3. The maximum number of cards which can be placed in each kanban.
4. The maximum number of running program for calculating error and STD.
5. The maximum number of iteration in each run.
6. The population size.
7. The minimum number cards which can be placed in each kanban.
8. The operation time for each product in total step.
9. The key of starting program for generating initial population.
The key of starting GA.
The key of starting GA-VNS.
The key of starting VNS.
The key of displaying efficient algorithm which is consist of error and STD/mean.
The member of initial population which is utilised as initial solution.
It is criteria for calculating digression of solution from optimal solution.
It is criteria for calculating digression of solution from average solution.
The key for implementing program step to step in order to capture costs.
The key for displaying terminal solution.
The table which is displayed initial and intermediate and terminal solution.

Afterward, results of GA, VNS and GA-VNS SMs are shown. The first feasible solution of VNS for four, five and six kanbans are shown in Tables 2, 5 and 8, respectively.

**Figure 6** Program form-1 (see online version for colours)

**Figure 7** Program form-2 (see online version for colours)
The first feasible solution of VNS algorithm for four kanbans

<table>
<thead>
<tr>
<th>Binary code</th>
<th>Kanban 1</th>
<th>Kanban 2</th>
<th>Kanban 3</th>
<th>Kanban 4</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010111...101</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>1,587</td>
</tr>
</tbody>
</table>

The first population of the GA and GA-VNS algorithm for four kanbans

<table>
<thead>
<tr>
<th>ID</th>
<th>Binary code</th>
<th>Kanban 1</th>
<th>Kanban 2</th>
<th>Kanban 3</th>
<th>Kanban 4</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1010111...101</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>1,587</td>
</tr>
<tr>
<td>2</td>
<td>110010...001</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1,412</td>
</tr>
<tr>
<td>3</td>
<td>100111...Ill</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1,467</td>
</tr>
<tr>
<td>4</td>
<td>111001...110</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1,287</td>
</tr>
<tr>
<td>5</td>
<td>1010111...101</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>1,421</td>
</tr>
<tr>
<td>6</td>
<td>110010...001</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>1,532</td>
</tr>
<tr>
<td>7</td>
<td>100111...Ill</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>1,301</td>
</tr>
<tr>
<td>8</td>
<td>111001...110</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>1,387</td>
</tr>
<tr>
<td>9</td>
<td>111001...110</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1,254</td>
</tr>
<tr>
<td>10</td>
<td>1010111...101</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>1,298</td>
</tr>
<tr>
<td>11</td>
<td>110010...001</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>1,312</td>
</tr>
<tr>
<td>12</td>
<td>100111...Ill</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1,365</td>
</tr>
<tr>
<td>13</td>
<td>100111...Ill</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1,055</td>
</tr>
<tr>
<td>14</td>
<td>111001...110</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1,301</td>
</tr>
<tr>
<td>15</td>
<td>111001...110</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1,543</td>
</tr>
<tr>
<td>16</td>
<td>1010111...101</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1,175</td>
</tr>
</tbody>
</table>

The first populations of the GA and GA-VNS for four, five and six kanbans are shown in Tables 3, 6 and 9, respectively.

To compute error (2) and standard deviation (STD) (2), it is sufficient to have the optimal solution of $i^{th}$ run of SM and the number of all runs during the simulation:

$$\text{error} = \sum_{i=1}^{\text{run number}} \frac{\text{optimal solution, } - \text{best optimal solution}}{\text{best optimal solution}}$$

$$\text{STD} = \sum_{i=1}^{\text{run number}} \frac{\text{optimal solution, } - \text{mean optimal solution}}{\text{run number}}$$

Calculating error and CV for four, five and six kanbans are shown in Tables 4, 7 and 10, respectively.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Best solution in five runs</th>
<th>Error</th>
<th>Std/mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>1111</td>
<td>0.07460</td>
<td>0.0856</td>
</tr>
<tr>
<td>VNS</td>
<td>1100</td>
<td>0.06150</td>
<td>0.0535</td>
</tr>
<tr>
<td>GA-VNS</td>
<td>1043</td>
<td>0.03456</td>
<td>0.0435</td>
</tr>
</tbody>
</table>
Table 5  The first feasible solution of the VNS algorithm for five kanbans

<table>
<thead>
<tr>
<th>Binary code</th>
<th>Kanban 1</th>
<th>Kanban 2</th>
<th>Kanban 3</th>
<th>Kanban 4</th>
<th>Kanban 5</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>10101111..101</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>1,687</td>
</tr>
</tbody>
</table>

Table 6  The first population of the GA and GA-VNS algorithm for five kanbans

<table>
<thead>
<tr>
<th>ID</th>
<th>Binary code</th>
<th>Kanban 1</th>
<th>Kanban 2</th>
<th>Kanban 3</th>
<th>Kanban 4</th>
<th>Kanban 5</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10101111..101</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>1,687</td>
</tr>
<tr>
<td>2</td>
<td>110010...001</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1,477</td>
</tr>
<tr>
<td>3</td>
<td>100111...111</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>1,567</td>
</tr>
<tr>
<td>4</td>
<td>111001...110</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1,343</td>
</tr>
<tr>
<td>5</td>
<td>10101111..101</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>1,569</td>
</tr>
<tr>
<td>6</td>
<td>110010...001</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1,376</td>
</tr>
<tr>
<td>7</td>
<td>100111...111</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>1,301</td>
</tr>
<tr>
<td>8</td>
<td>111001...110</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1,654</td>
</tr>
<tr>
<td>9</td>
<td>111001...110</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>1,578</td>
</tr>
<tr>
<td>10</td>
<td>10101111..101</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1,302</td>
</tr>
<tr>
<td>11</td>
<td>110010...001</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>5</td>
<td>1,398</td>
</tr>
<tr>
<td>12</td>
<td>100111...111</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1,287</td>
</tr>
<tr>
<td>13</td>
<td>100111...111</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>1,276</td>
</tr>
<tr>
<td>14</td>
<td>111001...110</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1,301</td>
</tr>
<tr>
<td>15</td>
<td>111001...110</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1,346</td>
</tr>
<tr>
<td>16</td>
<td>10101111..101</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1,367</td>
</tr>
</tbody>
</table>

Table 7  Calculating relative error and std/mean for five kanbans

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Best solution in five runs</th>
<th>Error</th>
<th>Std/mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>1334</td>
<td>0.0612</td>
<td>0.0756</td>
</tr>
<tr>
<td>VNS</td>
<td>1312</td>
<td>0.0598</td>
<td>0.0512</td>
</tr>
<tr>
<td>GA-VNS</td>
<td>1297</td>
<td>0.0235</td>
<td>0.0387</td>
</tr>
</tbody>
</table>

Table 8  The first feasible solution of the VNS algorithm for six kanbans

<table>
<thead>
<tr>
<th>Binary code</th>
<th>Kanban 1</th>
<th>Kanban 2</th>
<th>Kanban 3</th>
<th>Kanban 4</th>
<th>Kanban 5</th>
<th>Kanban 6</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>10101111..101</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>1,786</td>
</tr>
</tbody>
</table>

Table 9  The first population of the GA and GA-VNS algorithm for six kanbans

<table>
<thead>
<tr>
<th>ID</th>
<th>Binary code</th>
<th>Kanban 1</th>
<th>Kanban 2</th>
<th>Kanban 3</th>
<th>Kanban 4</th>
<th>Kanban 5</th>
<th>Kanban 6</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10101111..101</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>1,786</td>
</tr>
<tr>
<td>2</td>
<td>110010...001</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1,567</td>
</tr>
<tr>
<td>3</td>
<td>100111...111</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>1,634</td>
</tr>
<tr>
<td>4</td>
<td>111001...110</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1,497</td>
</tr>
<tr>
<td>5</td>
<td>10101111..101</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>1,634</td>
</tr>
<tr>
<td>6</td>
<td>110010...001</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1,745</td>
</tr>
</tbody>
</table>
Table 9  The first population of the GA and GA-VNS algorithm for six kanbans (continued)

<table>
<thead>
<tr>
<th>ID</th>
<th>Binary code</th>
<th>Kanban 1</th>
<th>Kanban 2</th>
<th>Kanban 3</th>
<th>Kanban 4</th>
<th>Kanban 5</th>
<th>Kanban 6</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>100111...Ill</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>1,423</td>
</tr>
<tr>
<td>8</td>
<td>111001...110</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1,734</td>
</tr>
<tr>
<td>9</td>
<td>111001...110</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1,687</td>
</tr>
<tr>
<td>10</td>
<td>1010111...101</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1,545</td>
</tr>
<tr>
<td>11</td>
<td>110010...001</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>1,567</td>
</tr>
<tr>
<td>12</td>
<td>100111...Ill</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1,356</td>
</tr>
<tr>
<td>13</td>
<td>100111...Ill</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>1,367</td>
</tr>
<tr>
<td>14</td>
<td>111001...110</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1,301</td>
</tr>
<tr>
<td>15</td>
<td>111001...110</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1,456</td>
</tr>
<tr>
<td>16</td>
<td>1010111...101</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1,365</td>
</tr>
</tbody>
</table>

Table 10  Calculation of error, standard deviation and mean for six kanbans

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Best solution in five runs</th>
<th>Error</th>
<th>Std/mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>1334</td>
<td>0.06120</td>
<td>0.0700</td>
</tr>
<tr>
<td>VNS</td>
<td>1312</td>
<td>0.05980</td>
<td>0.0462</td>
</tr>
<tr>
<td>GA-VNS</td>
<td>1297</td>
<td>0.03867</td>
<td>0.0356</td>
</tr>
</tbody>
</table>

Comparing relative error and critical values (CVs) which are standard deviation and mean (std/mean) between GA, VNS and GA-VNS are shown in Figures 8 and 9, respectively.

Figure 8  Comparison of relative error between GA, VNS and GA-VNS (see online version for colours)

Considering Figure 8, minimum errors has been occurred using GA-VNS model for all values of kanban (four, five and six). Meanwhile, maximum error belongs to GA model for all values of kanban. Putting all in one, minimum error has been obtained when GA-VNS model was used with five kanbans.
Figure 9  Comparison of standard deviation and mean between GA, VNS and GA-VNS  
(see online version for colours)

Examining Figure 9, minimum CV has been occurred using GA-VNS model for all 
values of kanban (four, five and six). Meanwhile, maximum CV belongs to GA model for 
all values of kanban. Putting all in one, minimum CV has been obtained when GA-VNS 
model was used with six kanbans.

According to above figures, GA-VNS can be used as a standard optimisation 
technique with any SM with respect to multi-stage SCS.

6 Conclusions

This paper has presented a GA, VNS and hybrid GA-VNS to optimise the SMs of 
multi-stage SCSs. It is used to solve batch size problems for kanbans in SCSs under JIT 
philosophy. An important issue in industrial and academic field is SCM. There are 
several methods for modelling, analysing and solving complex decision making problems 
in supply chains. There are a set of suppliers, manufacturers, warehouses and stores in 
SCM. Computer simulation is widely used for solving complex supply chain problem. 
Some parameters of SCS are stochastic. So, simulation is the best technique for 
modelling of this problem. Some parameters of SCS are stochastic. So, simulation is the 
best technique for modelling of this problem. Also, because the model has stochastic 
nature, hybrid GA-VNS is applied to solve optimisation and design problems. This paper 
is presented a new approach based on computer simulation, just in time, GA and VNS are 
presented for optimisation of these systems. Under JIT philosophy, kanbans are used for 
controlling objective of system. Finally, a proposed approach was applied for a numerical 
experiment. This study has been shown that GA-VNS model provides minimum relative 
error, mean and standard deviation. Also, it is claimed that GA-VNS can be used as a 
standard optimization technique with any SM with respect to multi-stage SCS.

GA is known many years ago and it is a very general algorithm and so will work well 
in any search space. All you need to know is what you need the solution to be able to do 
well, and a GA will be able to create a high quality solution. VNS, a met heuristic 
proposed just a few years ago is based upon a simple principle: systematic change of
neighbourhood within the search. According to the results of this paper, GA-VNS model provides minimum relative error, mean and standard deviation. Therefore, the proposed hybrid model can be used as a standard optimisation technique along with SM considering multi-stage SCS.

Table 11 presents the features of the proposed methodology. It helps manage multi-stage SCS controlled by kanban system. It is also capable of applying GA, VNS and hybrid GA-VNS to optimise SMs of multi-stage SCS.

Table 11  The features of proposed methodology versus other studies

<table>
<thead>
<tr>
<th>Method</th>
<th>Multi-stage SCS</th>
<th>Kanban systems</th>
<th>JIT production philosophy</th>
<th>SM GA</th>
<th>Variable neighbourhood search algorithm</th>
<th>Hybrid GA-VNS approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>This study</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Barkley (1991)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sarker and Balan (1999)</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wang and Sarker (2006)</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Azadeh et al. (2010a)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Azadeh et al. (2010)</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Moattar Husseini et al. (2006)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Acknowledgements

The authors are grateful for the valuable comments and suggestion from the respected reviewers. Their valuable comments and suggestions have enhanced the strength and significance of our paper. This study was supported by a grant from University of Tehran (Grant No. 8106013/1/14). The authors are grateful for the support provided by the College of Engineering, University of Tehran, Iran.

References


Optimisation of multi-stage supply chain systems


Appendix 1

Procedure genetic algorithm

- Generation initial population with size hundred
- For i <= number of iteration
  - Determine fitness function for each gene
  - Selecting best gene from four genes which are randomly selected. We define three different neighbor for each selected gen and performing cross over and motivation and generating of new population.
- End
- Announcing best solution
Appendix 2

Procedure variable neighbourhood search algorithm

Finding a initial solution by GA algorithm
Boolean variable 1 = yes.
Boolean variable 2 = yes.
Boolean variable 3 = yes.

* While Boolean variable 1 == yes
Selecting two member of sequence randomly and relocating with together.
If new fitness solution is better than before
Boolean variable 1 = yes
Best solution = new solution
Else
Boolean variable 1 = no
Boolean variable 2 = yes
End if
End while 1

While Boolean variable 2 == yes
Selecting four member of sequence randomly and relocating with together.
If new fitness solution is better than before
Boolean variable 1 = yes
Best solution = new solution
GOTO LINE *
Else
Boolean variable 2 = no
End if
End while

For i <= number of iteration
Selecting sex member of sequence randomly and relocating with together
If new fitness solution is better than before
Best solution = new solution
Boolean variable 1 = yes
GOTO LINE *
End if
End for

Announcing best solution
Appendix 3

Procedure GA-VNS algorithm

Generation population with size twenty
For i <= number of iteration 1
Determine fitness function for each gene
Selecting best gene from four genes which are randomly selected. We define three different
neighbors for each selected gene and performing cross over and motivation then one of three
generated sequences transmits to VNS algorithm *
(Start*)
While Boolean variable 1 == yes
Selecting two member of sequence randomly and relocating with together.
If new fitness solution is better than before
Boolean variable 1 = yes
Best solution = new solution
Else
Boolean variable 1 = no
Boolean variable 2 = yes
End
End
While Boolean variable 2 == yes
Selecting four member of sequence randomly and relocating with together.
If new fitness solution is better than before
Boolean variable 1 = yes
Best solution = new solution
GOTO LINE *
Else
Boolean variable 2 = no
End if
End while
For i <= number of iteration 2
Selecting sex member of sequence randomly and relocating with together
If new fitness solution is better than before
Best solution = new solution
Boolean variable 1 = yes
GOTO LINE *
End if
End for 2
(End*)
And generating of new population
Determine fitness function for each gene
End for 1
Announcing best
Appendix 4

Simulation models of GA, VNS and GA-VNS

Figure A1  GA simulation model

Figure A2  VNS simulation model
Figure A3  Hybrid VNS-GA simulation model