International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 21 (2017) pp. 11658-11667

On optimization of a Multi-Retailer Multi-Product VMI Model with Variable Replenishment Cycle

Debashish Kumar[1]
Research Scholar, Department of Computer Applications,
National Institute of Technology, Jamshedpur-831014, Jharkhand, India.

Danish Ali Khan
Associate Professor, Department of Computer Applications,
National Institute of Technology, Jamshedpur-831014, Jharkhand, India.

Tarni Mandal
Associate Professor, Department of Mathematics
National Institute of Technology, Jamshedpur-831014, Jharkhand, India.

[1]*Orcid Id: 0000-0002-4878-0953*

Abstract

This paper discusses how a vendor manages multiple products within its multiple non-identical retailers operating under a vendor managed inventory (VMI) contract that allows unequal shipment frequencies to the retailers. The vendor replenishment cycle for each product is taken to be an integer multiple of the retailer replenishment cycle where vendor is penalized for exceeding the upper stock limits at the retailer premises for each product. A mixed-integer nonlinear model is developed to address the replenishment problem among multiple retailers for multiple products that minimizes the joint relevant inventory costs under unequal shipment frequency. A particle swarm optimization (PSO) and simulated annealing (SA) algorithm along with a numerical illustration is also provided to demonstrate the proposed model. Results show that PSO algorithm finds optimal or near optimal solutions for the total system cost, and it proposes substantial savings compared to the solution obtained in SA and traditional optimizer software.

Keywords: supply chain, inventory model, vendor managed inventory, unequal shipment, penalty cost, particle swarm optimization, simulated annealing.

INTRODUCTION

Vendor managed inventory (VMI) is one of the coordination mechanism that has been gaining a lot of attention recently and many successful businesses have demonstrated the benefit of VMI, e.g. Wal-Mart, JC Penny, HP, Shell, Campbell Soup, Barilla, Johnson & Johnson, Kodak Canada Inc. and White Bread Beer [2, 9 & 39]. In VMI, retailer hands over the operational control of inventory within a mutually agreed framework and the performance targets are monitored and updated on as-needed basis [29], and the retailer must provide all relevant information like sales, inventory level etc. while the vendor determine the timing and replenishment quantity of the product for each retailer based on that information. In order to reap maximum benefit, the vendor always tends to move much of its inventory to the retailer premises by shipping large quantities, whereas each retailer restricts the vendor to keep inventory level below an agreed maximum level. Thus, it is quite common in VMI contracts that the retailer is protected from over-supplies from the vendor by means of a mutually agreed-upper stock limit at the retailer and the vendor is penalized for items exceeding these limits [4, 6 & 36]. Thus, retailer relieved of keeping track of its inventory and placing orders with the vendor from time to time, thereby eliminating its order costs [28].

Benefit of VMI adoption cited in literature include reduction in costs related to ordering [24], inventory [44], transportation [48], and improvement in different areas like production planning [9], service level [45], customer quality [49], forecasting accuracy [22] etc. VMI has some advantages not only for both the retailer and for the supplier, but also the customer service levels may increase in terms of the reliability of product availability [32]. Thus, VMI is best viewed as a part of a process of ongoing improvement, rather than as a standalone arrangement [13]. The customize replenishment frequencies in VMI is more beneficial when the supply chain retailers are heterogeneous in terms of their cost and demand parameters. Therefore, if the retailers are identical, then there is less need for VMI [17].

Many researchers have tried to model mathematically different aspects of VMI systems. Viswanathan and Piplani in 2001 [47] proposed a strategy where the vendor specifies common replenishment period in a single-vendor multi-retailer supply chain. Choi et al. in 2004 [5] presented an analytical model to measure the service level under VMI. Cetinkaya and Lee in 2004 [3] proposed a two-echelon dynamic lot-sizing model where demands had to be satisfied during pre-specified time windows with early and late delivery penalties. Darwish and Odah in 2010 [7] proposed a single-vendor multi-retailer VMI model in which the vendor incurs a penalty cost for items exceeding the bounds that are agreed upon in a contractual agreement between vendor and retailers. Mateen et al. in 2015 [28] developed an approximate model in which the vendor replenishes all the retailers at the same time under stochastic demand and in case of a shortage at the vendor, the available stock is allocated to the retailers on the basis of equal stock out probability. Mateen and Chatterjee in 2015 [27] proposed a detailed analysis of four different replenishment policies for single-vendor multi-retailer single-product under VMI contract.

Modern heuristic algorithms are considered as practical tools for nonlinear optimization problems. The population-based optimization approaches can find very good solution within the solution space efficiently and effectively, when solving the constraint np-hard problem. Meta-heuristic algorithms have proposed to solve some of the existing developed inventory models in the literature and have received an increasing attention from the researchers and practitioners since they give us an alternative to traditional optimization techniques. Some of these algorithms are: genetic algorithm [1, 8, 30, 40, 41], particle swarm optimization (PSO) algorithm [8, 16, 20, 26, 31, 33, 34 & 35], simulated annealing [23, 30 & 42], ant colony algorithm optimization [10] and harmony search [15, 25 & 40]. Most of the population-based search approaches like genetic algorithm are motivated by evaluation as seen in nature, while PSO is motivated from the simulation of social behavior. Nevertheless, they all work in the same way so that the individuals of the population can be expected to move towards better solution space [37]. Another widely used meta-heuristic algorithm that has been used to solve sophisticated problems in different fields of study is the simulated annealing (SA) algorithm [23 & 42].

Under VMI contract, many authors have tried to come up with single-vendor single-retailer or single-vendor multi-retailer replenishment schemes for single product, and/or single-vendor single-retailer multi-product with constraint like order quantity, space, budget, product life time, maximum backorder level etc.

In this paper, we consider a single-vendor multi-retailer multi-product supply chain operating under VMI contract with unequal shipment frequencies. As suggested by the extant literature on VMI, to overcome the over-supply problem at the retailer end, we consider an upper stock limit for each product at the retailer and penalty for the vendor for over-supply. In this case, the vendor incurs the storage cost for the extra quantity for each product beyond the retailers' storage capacity, which does not include the capital cost as the stock is owned by the retailer [17] and vendor is penalized whenever replenishment exceed this upper stock limit, which may be linked to the space constraint at the retailer [4, 6, 7, 14 & 36]. A mixed-integer nonlinear model is developed to address the replenishment problem among multiple retailers for multiple products that minimizes the joint relevant inventory costs under unequal shipment frequency and solve this hard optimization problem through a valuable and elegant approach based on PSO and SA algorithms.

The remainder of the paper is organized as follows: The next section presents a problem statement, associated notations and mathematical model developed for evaluating VMI systems. In order to demonstrate the application of the proposed methodology, a solution procedure including particle swam optimization and simulated annealing algorithm, and a numerical illustration in following sections and the last section concludes the paper and recommendations for future research.

PROBLEM STATEMENT AND FORMULATION

Consider a supply chain where a vendor supplies n products to k retailers under a VMI agreement. The vendor orders the product from an external source with unlimited capacity. The consumer demand for each product at each retailer is known and constant, which must be meet without lost sales or backordering, while each retailer faces independent, stationary and normally distributed demand for each product. The vendor replenishment cycle is taken to be an integer multiple of the retailer replenishment cycle where vendor is penalized for exceeding the upper stock limits of each product at the retailer premises. In such a situation, the vendor is responsible for initiating the orders and setting the replenishment quantities on behalf of the retailers.

The objective of this study is to determine the optimal replenishment policy with unequal shipment frequencies under VMI agreement that is economically beneficial to the entire supply chain. It is shown in Figure 1 that the variation over time of the stock of i^{th} product at each echelon is saw-toothed in shape, which simplifies the formulation of the total holding costs per unit of time for both echelons. Moreover, the conditions under which the vendor operates are as follows:

- There is no procurement lead time for all products for the vendor.
- Vendor orders the products from an external source having unlimited supply.
- Cost of holding one unit of each product per unit of time at the vendor facility is less than that of each one of the retailers.
- Each retailer specifies an upper stock level for each product under VMI, and the vendor penalize financially whenever that level exceeded.
- Retailers face independent, stationary and normal distributed demand for each product.
- There are no shortages in retailer's end.

Retailers bear only their own holding cost and all the remaining costs, viz. vendor order and holding costs, retailer order costs, and penalty costs for each product are borne by the vendor.

Notations

Let n be the number of products, and i be the index for products, $i = 1, 2, \ldots n$, k be the number of retailers, and j be the index for retailers, $j = 1, 2, \ldots k$, the following notations will be used to develop the proposed model:

- A_i^v Vendor's fixed ordering cost per order for i^{th} product ($/order)
- A_{ij}^r Cost charged for i^{th} product to the j^{th} retailer for receiving its ordered shipment and cost incurred by the vendor for initiating and releasing an order for i^{th} product to the j^{th} retailer ($/order)
- D_i^v $\sum_{j=1}^{k} D_{ij}$, total demand to be supplied by the vendor for i^{th} product (unit/order)
- D_{ij}^r Demand rate per unit of time for i^{th} product to j^{th} retailer (unit/order)
- h_i^v vendor holding cost per unit per unit of time for i^{th} product ($/unit/order)
- h_{ij}^r j^{th} retailer holding cost per unit per unit of time for i^{th} product ($/unit/order)
- π_{ij} Penalty cost per unit for the over-stock quantity for i^{th} product at the j^{th} retailer ($/order)
- U_{ij} Upper stock limit for i^{th} product at the j^{th} retailer facility (units)
- T_i^v Vendor reorder cycle for i^{th} product (years) [a decision variable]
- T_{ij}^r replenishment cycle for i^{th} product to j^{th} retailer (years)
- m_{ij} number of deliveries made during the vendor's cycle T_i^v for i^{th} product to the j^{th} retailer [a decision variable]
- q_{ij} $= T_{ij}^r D_{ij}^r$, ordering quantity of i^{th} product shipped to the j^{th} retailer (units) [a decision variable]
- z_{ij} over-stock quantity for i^{th} product at the j^{th} retailer (units)

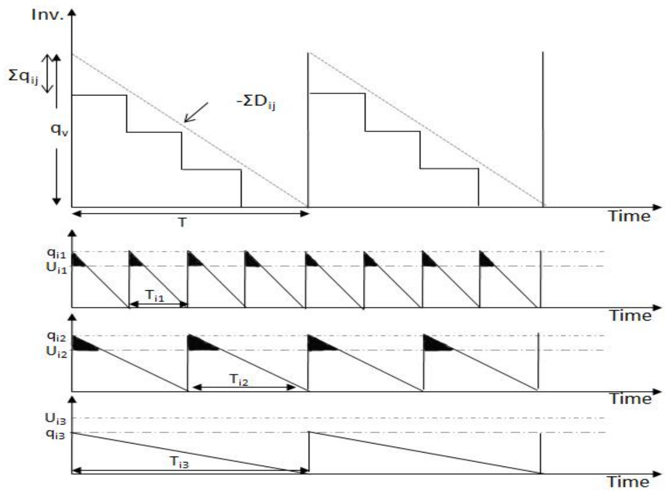

Figure 1: Demand and shipment variation for i^{th} product over time of the inventory levels for one vendor and three retailers.

Mathematical Model

In spite of Hariga et al. [17], Hariga et al. [18] and Verma et al. [46] who extended the single-vendor multi-retailer single-product supply chain problem with capacity constraints of Darwish and Odah [7] to considered unequal shipment frequencies, in this paper, a model for the single-vendor multi-retailer multi-product supply chain problem is developed. To do this, a model for a single-vendor multi-retailer developed first for one product and then it will be extended to include single-vendor, multiple retailers, and several products.

Model for i^{th} product and for j^{th} retailer:

Vendor annual order costs = $\frac{A_i^v}{T_i^v}$

Retailer annual order cost = $\frac{A_{ij}^r}{T_{ij}^r}$

Average annual holding cost for the i^{th} product at the j^{th} retailer = $\frac{1}{2} D_{ij}^r T_{ij}^r (h_{ij}^r - h_i^v)$

Average holding cost for the vendor for the i^{th} product

$$= \frac{1}{2} D_i^v T_i^v h_i^v$$

Referring to Figure 1, which shows the variation over time of the vendor inventory levels and three retailers for the i^{th} product, the over stock for the i^{th} product at the j^{th} retailer, z_{ij} can be written as:

$$z_{ij} = T_{ij}^r D_{ij}^r - U_{ij}, \quad j \in R$$

$$z_{ij} = 0, \quad j \in \bar{R}$$

Where R is the set of retailers with over stock and $\bar{R}$ is the complement set of R.

Since the penalty cost is charged per unit of time and based on the average over-stock (the shaded triangles in Fig. 1). Therefore, the total over-stock penalty costs per unit of time for i^{th} item can be written as [7, 17 & 18]:

$$\frac{\sum_{j=1}^{k} \pi_{ij} m_{ij} \frac{z_{ij}^2}{2D_{ij}^r}}{T_i^v}, \textit{or after substituting } T_i^v = m_{ij} T_{ij}^r \textit{, we have}$$

$$\sum_{j=1}^{k} \pi_{ij} \frac{z_{ij}^2}{2T_{ij}^r D_{ij}^r}$$

subject to,

$$T_{ij}^r D_{ij}^r - U_{ij} \le z_{ij} \quad \textit{for } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, k$$

$$z_{ij} \ge 0 \quad \textit{for } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, k$$

The optimization problem for n products and k retailers can then be stated as

$$Min\ TC = \sum_{i=1}^{n} \frac{A_i^v}{T_i^v} + \sum_{i=1}^{n}\sum_{j=1}^{k} \frac{A_{ij}^r}{T_{ij}^r} + \sum_{i=1}^{n} \frac{1}{2} D_i^v T_i^v h_i^v + \sum_{i=1}^{n}\sum_{j=1}^{k} \frac{1}{2} D_{ij}^r T_{ij}^r (h_{ij}^r - h_i^v) + \sum_{i=1}^{n}\sum_{j=1}^{k} \pi_{ij} \frac{z_{ij}^2}{2T_{ij}^r D_{ij}^r}$$

subject to,

$$T_{ij}^r D_{ij}^r - U_{ij} \le z_{ij} \quad for\ i = 1,2,\dots,n \text{ and } j = 1,2,\dots,k$$
$$z_{ij} \ge 0 \quad for\ i = 1,2,\dots,n \text{ and } j = 1,2,\dots,k$$
$$T_{ij}^r \ge 0 \quad for\ i = 1,2,\dots,n \text{ and } j = 1,2,\dots,k \qquad (1)$$
$$T_i^v = m_{ij} T_{ij}^r \quad for\ i = 1,2,\dots,n \text{ and } j = 1,2,\dots,k$$
$$m_{ij} = \{1,2,\dots.\} \quad for\ i = 1,2,\dots,n \text{ and } j = 1,2,\dots.,k$$

THE SOLUTION PROCEDURE

The model that was formulated in the previous section belongs to mixed-integer nonlinear programming (MINLP) problem. The goal is to determine the vendor order cycle (T_i^v), number of deliveries to each retailer for each product (m_{ij}) and order up to level to each retailer for each product (q_{ij}) so that the total cost (TC) under VMI policy given in (1) is minimized and all constraints are fulfilled. The difficulty originated from the nonlinearity of the objective function and constraints, and integer restriction of m_{ij} in the MINLP formulation. Since the model (1) is mixed-integer nonlinear programming (MINLP) in nature an analytical solution (if any) to the problem is difficult. In addition, efficient treatment of mixed-integer nonlinear optimization is one of the most difficult problems in practical optimization [12] and one cannot use an exact algorithm to solve them. Hence, typically some non-precise heuristic search algorithms are needed for a near-optimum solution in shorter period. Furthermore, a meta-heuristic algorithm is a heuristic method to solve a very general class of computational problem by combining user-given black box procedure (usually heuristic themselves) in the hope of obtaining a more robust procedure.

Many researchers have successfully implemented meta-heuristic approaches to solve complicated optimization problem in various fields of scientific and engineering disciplines. Some of these algorithms such as particle swarm, ant colony optimization, genetic algorithm and simulating annealing have been employed to solve complex supply chain problems. However, the particle swarm optimization (PSO) search algorithm, due to its good performance in solving complex optimization problems [20, 33 & 42], is model in (1). In addition, to validate the results and to compare its performance, a simulated annealing (SA) algorithm also employed to solve the problem as well [30 & 41]. Further, since no benchmarks can be found in the literature to assess the performance of PSO and SA, a branch and bound solver based programming in LINGO software is employed to solve the model (1) as well.

Particle swarm optimization

Particle swarm optimization (PSO) is a population based stochastic optimization techniques proposed by Kennedy and Eberhart [21], inspired by social behaviour of bird flocking or fish schooling and based on evolutionary computational techniques where system is initialized with a population of random solution and searches for optima by updating generations. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimal particles.

The type of particle is associated with the number of decision variables involved in a problem. In this paper, the decision variables of the proposed model are vendor's product wise common replenishment cycle (T_i), number of deliveries for each product to each retailer in each vendor order cycle (m_{ij}) and order quantity (q_{ij}) to respective retailers. The particle is modelled by a $1 * (n + 2(n * k))$ matrix for n products and k retailers, and so, for a 5 products and 4 retailers case, the particle is represented as 1x45 matrix. The general form of particle is considered as $[T_1\ T_2 \dots T_5\ m_{11}\ m_{12} \dots m_{54}\ q_{11}\ q_{12} \dots q_{54}]$.

As PSO is a population based optimization algorithm, each particle is an individual and swarm is composed of particles. Mapping between swarm and particles in PSO is similar to the relationship between population and chromosomes in genetic algorithm [43].

PSO algorithm start with initialization of random particles position (solution) and exploration velocities, and then searches for optima by updating exploration velocities and then position update of particles. In every iteration, each particle is updated by following two “best” values. The first one (p_{best}) is the best solution (fitness) achieved so far. Another best value (g_{best}) is a global best solution tracked by the particle swarm optimizer and obtained so far by any particle in the population. When a particle takes part of the population as its topological neighbours, the best value is a local best (l_{best}).

The initialization process allows the swarm particles and exploration velocity to be generated under feasible solution space as given in equation (2) and (3) assuming constant time interval to be 1.

$$X_0^i = X_{min}^i + rand() * (X_{max}^i - X_{min}^i) \qquad (2)$$

$$V_0^i = position/time = X_0^i \qquad (3)$$

Where X_{min}^i and X_{max}^i are the lower and upper bounds on the design variables values respectively. X_0^i and V_o^i are the position and velocity of the i^{th} particle in the population at

0^{th} iteration respectively, and $rand()$ is a uniformly distributed random number between [0, 1].

In each step, the feasibility of a generated solution is checked and if a solution vector does not satisfy a constraint, the related vector solution will be punished by a big penalty on its fitness.

In a d-dimension searching space and a swarm of N particle searching for the global optimal solution within the searching space. Three d-dimensional vectors are assigned to each particles and the k^{th} particle: position, denoted by $X_{kd} = (X_{k1}, X_{k2}, \dots, X_{kd})$; the rate of change of position of particle, also called velocity, denoted by $V_{kd} = (V_{k1}, V_{k2}, \dots, V_{kd})$; and best personal position, denoted by $P_{kd} = (P_{k1}, P_{k2}, \dots, P_{kd})$. Where the velocity vector represents the distance a particle will traverse in each dimension in each iteration of the algorithm and the best personal position vector is the best visited position for a particle. Particles also need to be aware of the best position visited by the whole swarm, which is denoted by $G_{kd} = (G_{k1}, G_{k2}, \dots, G_{kd})$.

Step 1: Initialize the particle position $\{X_{11}$, k=1,2,…N} and exploration velocity $\{V_{11}$, d=1,2,…D}, where 'k' denote the number of particles, and 'd' denotes maximum number of dimension within minimum $[X_{min}]$ and maximum $[X_{max}]$ limits for each dimension.
Step 2: Set $P_{11} = X_{11}, V_{max} = 0.5 * X_{max}, P_{best} = X_{kd}$ and $G_{best} = X_{11}$
Step 3: Set $k = 1$ and $d = 1$
Step 4: if $f(X_{kd}) < f(P_{kd})$ then $P_{kd} = X_{kd}$
else $G_{kd} = min(P_{kd})$
Step 5: calculate new velocity (V_{kd}^{new}) given in equation (4)
Step 6: if $V_{kd}^{new} > V_{max}$ then $V_{kd} = V_{max}$
Step 7: $X_{kd}^{new} = X_{kd} + V_{kd}^{new}$
Step 8: set $k = k + 1$
Step 9: if $k < N$ go to step 5
else $d = d + 1$
Step 10: if $d < D$ then go to step 5
Step 11: Print G_{kd} and $f(G_{kd})$
Stop

Figure 2: General structure of proposed PSO algorithm

Then to update velocity and position vectors of the particles for the next iteration, the following equations (4) and (5) will be used respectively:

$$V_{kd}^{new} = w_{kd} + c1 * [rand1 * (P_{kd} - X_{kd})] + c2 * [rand2 * (G_{kd} - X_{kd})] \quad (4)$$

$$X_{kd}^{new} = X_{kd} + V_{kd}^{new} \quad (5)$$

Where w_{kd} is the inertia weight introduced by Shi and Eberhart [38]; $c1$ and $c2$ are the cognitive and social factors respectively; k represent the iteration number and $rand1$ and $rand2$ are the random number between [0, 1].

In PSO, the inertia weight (w_{kd}) is brought in for balancing the global and local search. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. By linearly decreasing the inertia weight from relatively large value to a small value, the PSO tends to have more global search ability at the beginning of the run while having more local search ability near end of the run [49]. The linear distribution of inertia weight is expressed as follows [31]:

$$w_{kd} = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} * iter \quad (6)$$

Where w_{max} is the initial weight; w_{min} is the final weight; $iter_{max}$ is the maximum iteration number and $iter$ is the current iteration number.

Figure 2 shows the general flowchart in the optimization of the proposed VMI model using PSO algorithm.

The simulation result presented by [37] illustrate that an inertia weight starting with a value close to 1 and linearly decreasing to 0.4 through the course of the run will give the PSO the best performance compare with all fixed inertia weight setting. In this research, a linearly decreasing inertia weight is used which starts at 0.9 (w_{max}) and ends at 0.4 (w_{min}), with $c1 = 2$, $c2 = 2$ and N=200.

Simulated Annealing

Simulated annealing (SA) is a probabilistic single-solution based search method inspired by the annealing process in metallurgy. In SA, a simulation is carried out of physical/chemical process, where annealing referred to as tempering certain alloys of metal or glass by heating above its melting point, holding its temperature, and then cooling it very slowly until it solidifies or produces high-quality materials. SA is basically composed of two stochastic processes: generation of solutions and acceptance of solutions. The generated temperature is responsible for the correlation between generated probing solutions and the original solution [11]. SA is best for solving unconstrained and bound-constrained optimization, but by introducing penalty method to handle the constraints, which penalizes infeasible points and as the temperature decreases, the algorithm reduces the extent of its search to converge to a minimum.

The SA involves four control parameters $\propto$ (the cooling schedule), T_0 (the initial temperature), the iteration of the internal loop, and maximum iteration of external loop. The cooling schedule for T is critical to efficiency of SA. If T is reduced too rapidly, a premature convergence to a local minimum may occur and if it is too slow, the algorithm is very slow to converge. Given a sufficiently large number of iteration at each temperature, SA is proved to converge almost surely to the global optimum. Fig. 3 presents the general framework of SA procedure for the proposed model (1) and the SA parameters best values are summarized in table 3 [2 & 19].

Step 1: Initialize temperature (T_0), random starting point (x_0) and other system parameters

Step 2: initialize $k = 0$, $f_{min} = f(x_0)$, $f_{new} = f(x_0)$, $f_{old} = f(x_0)$ and $x_{best} = x_0$

Step 3: while $T > 1e-10$ *or* $k < 2500$

$k = k + 1$

$i = 0$

while $i \leq 500$

calculate new neighbourhood points (x_{new})

calculate $f_{new} = f(x_{new})$

$\Delta = f_{new} - f_{old}$

if $\Delta < 0$ *then* $x_{best} = x_{new}$; $f_{old} = f(x_{new})$

if $\Delta > 0$ *and* $exp(-{}^{\Delta}/_{(k*T)}) > rand()$ *then* $x_{best} = x_{new}$; $f_{old} = f(x_{new})$

$i = i + 1$

$f_{min} = f_{old}$

end while

$T = T_0 * \propto^k$

end while

Step 4: output the final state x_{best} *and* f_{min}

stop

Figure 3: General structure of proposed SA algorithm

Constraint handling

The constrained problem is transformed into a sequence of unconstrained problems by adding penalty terms for each constraint violation, if a constraint is violated at any point, the objective function is penalized by an amount depending on the extent of constraint violation. Here the exterior penalty method is used to handle the constraints of the problem. This kind of penalty method penalizes infeasible points but does not penalize feasible points. In these methods, every sequence of unconstrained optimization finds an improved yet infeasible solution [20].

Table 1: The vendor's common product data.

	Product				
	1	2	3	4	5
D_i	1200	1200	1200	1200	1200
A_i	100	100	100	100	100
h_i	0.2	0.2	0.2	0.2	0.2

This penalty parameter approach is a popular constraint handling strategy. Thereafter, all constraint violations are added together to get the overall constraint violation which is denoted by the 'Ω' called penalty term.

$$\Omega(x^i) = \sum_{k=1}^{n} \omega^{(x^i)}$$

This constraint violation is then multiplied with a penalty parameter R_m and the product is added to each of the objective function values:

$$F_m(x^i) = f_m(x^i) + R_m\Omega(x^i)$$

The function F_m takes into account the constraint violations. For a feasible solution, the corresponding Ω term is zero and F_m becomes equal to the original objective function f_m. However, for an infeasible solution, $F_m > f_m$, thereby adding a penalty corresponding to total constraint violation.

A numerical illustration

The MINLP for the replenishment problem among multiple retailers for multiple products with unequal replenishment cycle developed in the preceding section will now be illustrated by a numerical example to show the effectiveness of the model. In order to show that the PSO algorithm is an effective means of solving the complicated mixed-integer np-hard model, the simulated annealing approaches of Taleizadeh et al. [42] and traditional optimization software approaches of Mateen et al. [27] are also employed to solve the numerical example.

Table 2: The retailer's common product data.

		Retailers			
		1	2	3	4
D_{ij}					
Product	1	150	250	350	650
	2	125	225	325	625
	3	100	200	300	600
	4	75	175	275	575
	5	50	150	250	550
A_{ij}					
Product	1	5	4	3	2
	2	5	4	3	2
	3	5	4	3	2
	4	5	4	3	2
	5	5	4	3	2
h_{ij}					
Product	1	0.6	0.5	0.4	0.3
	2	0.6	0.5	0.4	0.3
	3	0.6	0.5	0.4	0.3
	4	0.6	0.5	0.4	0.3
	5	0.6	0.5	0.4	0.3
U_{ij}					
Product	1	14	21	28	42
	2	12	18	24	36
	3	10	15	20	30
	4	8	12	16	24
	5	6	9	12	18
π_{ij}					
Product	1	1.5	2	1	1
	2	1.5	2	1	1
	3	1.5	2	1	1
	4	1.5	2	1	1
	5	1.5	2	1	1

We consider a system with a vendor supplying 5 products to 4 retailers under VMI contract where vendor penalized for oversupply of the products beyond the pre-defined product wise upper stock limit in retailer premises. The vendor and retailers' general data are given in Table 1 and 2 respectively, which derived from illustrative example proposed in Mateen et al. [28] for 5 products in equal proportion.

Table 3: The initial parameter values for PSO and SA

SA	PSO
Initial temperature (T_0) = 100	Inertia factor (w) = [0.4, 0.9]
Cooling factor (α) = 0.9	Self-confidence factor (C_1) = 2 Swarm confidence factor (C_2) = 2 Swarm size (N)=150
Stopping criteria (MaxIteration) = 2500 or $T < 1e-10$	Stopping criteria = N

The developed PSO and SA are coded in MATLAB R2017a software and also the proposed model is programmed in LINGO 16.0 software. The test problem solved on Intel Core™ i5 2.5 GHz with 4 GB RAM CPU. Table 3 shows the best combination of different values of the PSO and SA parameters used to obtain the solution. The best results obtained by the three methods are given in Table 4, 5 and 6. A comparison of the results of all three methods and percentage of improvement with respect to objective function values are given in Table 7, which shows that the PSO algorithm performs better than the SA algorithm and LINGO solution in terms of system total costs. Furthermore, the convergence path of the best results of the objective function for PSO and SA are shown in Figure 4 and 5 respectively.

Table 4: The best result of the PSO

Product i	Retailers m_{ij}	q_{ij}	Cycle Time T_i	Penalty Cost	Chain total cost (TC)
1	8, 7, 6, 8	15, 29, 47, 66	0.81101	6.699	1497.527
2	9, 9, 5, 5	12, 22, 58, 111	0.88735	7.566	
3	2, 6, 9, 7	39, 26, 26, 67	0.77950	11.264	
4	5, 6, 5, 5	14, 27, 51, 107	0.93383	12.402	
5	6, 5, 6, 6	7, 25, 35, 78	0.84839	6.141	

CONCLUSION AND RECOMMENDTION FOR FUTURE RESEARCH

In this paper, we have presented PSO and SA algorithms to solve a single-vendor multiple-retailer multiple-product vendor managed inventory system with unequal shipment frequencies. Results shows that the PSO algorithm always obtained better solutions than the solutions of the SA algorithm and LINGO software. Additionally, the PSO algorithm is simple and find near-optimal solutions in a very short time. However, constraint handling in SA is a challenging task and possible by introducing penalty parameter approach. Finally, we conclude that the PSO algorithm performs very well compare to SA algorithm.

Table 5: The best result of the SA

Product i	Retailers m_{ij}	q_{ij}	Cycle Time T_i	Penalty Cost	Chain total cost (TC)
1	5, 6, 5, 8	30, 42, 71, 82	1.01476	30.088	1525.912
2	5, 5, 8, 6	23, 42, 38, 98	0.93749	29.414	
3	6, 8, 7, 7	13, 20, 35, 69	0.80871	24.179	
4	6, 6, 7, 8	10, 24, 32, 59	0.81995	29.289	
5	7, 8, 6, 8	6, 17, 38, 62	0.90707	34.211	

Table 6: The best result of the LINGO

Product i	Retailers m_{ij}	q_{ij}	Cycle Time T_i	Penalty Cost	Chain total cost (TC)
1	4, 6, 7, 8	36, 40, 48, 78	0.960	42.893	1777.012
2	4, 6, 8, 10	30, 36, 39, 60	0.960	28.111	
3	4, 5, 6, 10	25, 40, 50, 60	1.0	52.251	
4	3, 6, 6, 6	18, 21, 33, 69	0.720	34.844	
5	3, 6, 6, 11	16, 24, 40, 48	0.960	38.345	

Table 7: The chain total cost obtained by the algorithms

Total cost			PSO vs SA Improvement (%)	PSO vs LINGO Improvement (%)
LINGO	SA	PSO		
1777.012	1525.912	1497.527	1.861	15.728

This study can be extended in many directions. An obvious extension is to consider the case of stochastic demand or where lead time varies with time or randomness is introduced into the demand pattern. Considering the retailers upper limit as decision variables is another possible extension. Different meta-heuristic search algorithms like genetic algorithm, ant colony optimization, differential evolution etc. may also be employed to solve the proposed model and a comparison may be made among the algorithms. The model can further extend to some practical situations, such as VMI model for multi-vendor multi-retailer multi-product case. We will consider these problems in the near future.

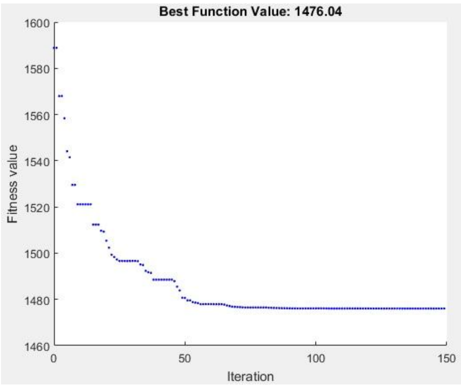

Figure 4: Convergence path of the objective function by PSO

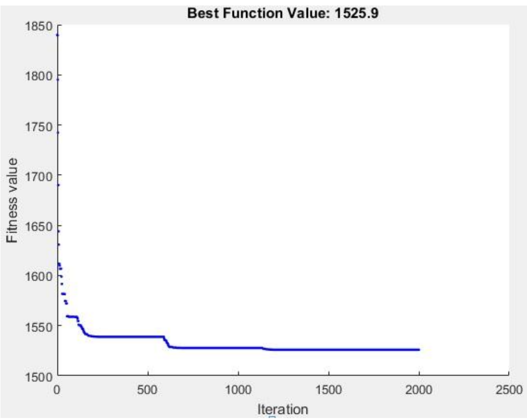

Figure 5: Convergence path of the objective function by SA

REFERENCES

[1] Al-Tabtabai H. and Alex P., "Using genetic algorithms to solve optimization problems in construction", Eng. Conf. Arch. Man., 6, pp. 121-132, 1999.

[2] Bouleimen K. and Lecocq H., "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version", European Journal of Operational Research, 149, pp. 268-281, 2003.

[3] Cetinkaya S. and Lee C.Y., "Stock replenishment and shipment scheduling for vendor-managed inventory systems", Management Sciences, 46(2), pp. 217-232, 2004.

[4] Chen X., Chen J. and Chen Y.F., "A coordination mechanism for a supply chain with demand information updating", International Journal of Production Economics, 103(1), pp. 347-361, 2006.

[5] Choi K.S., Dai J.G. and Song J.S., "On measuring supplier performance under vendor-managed-inventory programs in capacitated supply chains", Manufacturing & Service Operations Management, 6(1), pp. 53-72, 2004.

[6] Danese P., "The extended VMI for coordinating the whole supply network", Journal of Manufacturing Technology Management, 17(7), pp. 888-907, 2006.

[7] Darwish M.A. and Odah O.M., "Vendor-managed inventory model for single-vendor single-retailer supply chains", European Journal of Operation Research, 204(3), pp. 473-484, 2010.

[8] Dayal P.A.S., Raju G.S.N. and Mishra S., "Pattern synthesis using real coded genetic algorithm and accelerated particle swarm optimization", International Journal of Applied Engineering Research, 11(6), pp. 3753-3760, 2016.

[9] Dong Y. and Xu K., "A supply chain model of vendor managed inventory", Transportation Research Part E, 38, pp. 75-95, 2002.

[10] Dorigo M. and Stuzle T., "Ant Colony Optimization", MIT Press, Cambridge., 2004.

[11] Du Ke-Lin and Swamy M.N.S., "Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature", Birkhauser, 1st ed. ISBN - 9783319411910, 2016.

[12] El-Sharkawi L., "Modern heuristic optimization techniques", 1st edn., Wiley Inter. Science, New Jersey, 2008.

[13] Elvander M.S., Sarpola S. and Mattsson S., "Framework for characterizing the design of VMI systems", International Journal of Physical Distribution and Logistics Management, 37(10), pp. 782-798, 2007.

[14] Fry M.J., Kapuscinski R. and Olsen T.L., "Production and delivery under a (z, Z)-type vendor-managed inventory contract", Manufacturing and Service Operations Management, 3(2), pp. 151-173, 2001.

[15] Geem Z.W., Kim J.H. and Loganathan G.V., "A new heuristic optimization algorithm: harmony search", Simulation, 76, pp. 60-68, 2001.

[16] Guo Y.W., Li W.D., Mileham A.R. and Owen G.W., "Optimization of integrated process planning and scheduling using a particle swarm optimization approach", Int. j. Prod. Res., 40, pp. 1-22, 2008.

[17] Hariga M., Gumus M. and Daghfous A., "Storage constrained vendor managed inventory models with unequal shipment frequencies" Omega, 48, pp. 94-106, 2014.

[18] Hariga M., Gumus M., Daghfous A. and Goyal S.K., "A vendor managed inventory model under contractual storage agreement" Computers & Operations Research, 40, pp. 2138-2144, 2013.

[19] Ingber L., "Very fast simulated re-annealing", Mathematical Computational Modelling, 12 (8), pp. 967-973, 1989.

[20] Kadadevaramarth R.S., Chen J.C.H., Shankar B.L. and Rameshkumar K., "Application of particle swarm intelligence algorithm in supply chain network architecture optimization", Expert System and Applications, 39, pp. 10160-10179, 2012.

[21] Kennedy J. and Eberhart R., Particle swarm optimization", Proc. Of IEEE International Conference on Neural Networks (ICNN), Perth, Australia, Vol. IV, pp. 1942-1948, 1995.

[22] Khadar S.A., "VMI program improves forecasting & supply chain - Arasco's case study", The Journal of Business Forecasting, 26(3), pp. 29-32, 2007.

[23] Kirkpatrick S., Gellat C.D., Vecchi M.P., "Optimization by simulated annealing", Sciences, 220, pp. 671-680, 1983.

[24] Kumar P. and Kumar M., "Vendor Managed Inventory in retail Industry", White Paper, Tata Consultancy Services, 2003.

[25] Lee K.S. and Greem Z.W., "A new structural optimization method based on the harmony search algorithm", Comp. Struc., 82, pp. 781-798, 2004.

[26] Liu B., Wang L., Jin Y., "An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers", Comp. Op. Res., 35, pp. 2791-2806, 2008.

[27] Mateen A. and Chatterjee A.K., "Vendor Managed Inventory for single-vendor multi-retailer supply chains", Decision Support Systems, 70, pp. 31-41, 2015.

[28] Mateen A., Chatterjee A.K. and Mitra S., "VMI for single-vendor multi-retailer supply chain under stochastic demand" Computers & Industrial Engineering, 79, pp. 95-102, 2015.

[29] Michaelraj L.A. and Shahabudeen P., "Replenishment policies for sustainable business development in a credit based vendor managed inventory distribution system", Computers & Industrial Engineering, 56(1), pp. 260-266, 2009.

[30] Mousavi A.M., Hajipour V., Niaki S.T.A. and Alikar N., "Optimization multi-item multi-product inventory control system with discounted cash flow and inflation: Two calibrated meta-heuristic algorithms", Applied Mathematical Modelling, 37, pp. 2241-2256, 2013.

[31] Naka S., Genji T., Yura T. and Fukuyama Y., "Particle distribution state estimation using hybrid particle swarm optimization", IEEE Conference, pp. 815-820, 2001.

[32] Pasandideh S.H.R., Niaki S.T.A. and Nia A.R., "A genetic algorithm for vendor managed inventory control system of multi-product multi-constraint economic order quantity model", Expert Systems with Applications, 38, pp. 2708-2716, 2011.

[33] Priyadarshini K. and Moni Jackuline, "Analysis of block matching algorithm based on particle swarm optimization and differential evolution", International Journal of Applied Engineering Research, 11(3), pp. 2055-2058, 2016.

[34] Rahimi-Vahed A.R., Mirghorbani S.M. and Rabbani M., "A hybrid multi-objective particle swarm algorithm for a mixed-model assembly line sequencing problem", Engineering Optimization, 39(8), pp. 877-898, 2007.

[35] Sha D.Y., Hsu C., "A new particle swarm optimization for the open shop scheduling problem" Comp. Op. Res., 35, pp. 3243-3261, 2008.

[36] Shah J. and Goh M., "Setting operating policies for supply hubs", International Journal of Production Economics, 100 (2), pp. 239-252, 2006.

[37] Shi Y. and Eberhart R.C., "Parameter selection in particle swarm optimization", Annual Conference on Evolutionary Programming, San Diego, pp. 591-600, March 1998.

[38] Shi Y. and Eberhart R.C., "A modified particle swarm optimizer", IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

[39] Smaros J., Lehtonen J., Appelqvist P. and Holmstrom J., "The impact of increasing demand visibility on production and inventory control efficiency", International Journal of Physical Distribution & Logistics Management, 33(4), pp. 336-352, 2003.

[40] Taleizadeh A.A., Moghadasi H., Niaki S.T.A. and Eftekhari A., "An EOQ-joint replenishment policy to supply expensive imported raw materials with payment in advance", J. Appl. Sci., 8, pp. 4263-4273, 2008.

[41] Taleizadeh A.A., Niaki S.T.A. and Aryamezhad M.B., "Multi-product multi-constraint inventory control systems with stochastic period length and total discount under fuzzy purchasing price and holding costs", American Journal of Applied Sciences, 6(1), pp. 1-12, 2009.

[42] Taleizadeh A.A., Niaki S.T.A. and Hosseini V, "The multi-product multi-constraint newsboy problem with incremental discount and batch order", American Journal of Applied Sciences, 1, pp. 110-122, 2008.

[43] Tyan J. and Wee H. M., "Vendor managed inventory: a survey of the Taiwanese grocery industry", Journal of Purchasing and Supply Management, 9(1), pp. 11-18, 2003.

[44] Vergin R.C. and Barr K., "Building competitiveness in grocery supply through continuous replenishment planning: insight from the field", Industrial Marketing Management, 28, pp. 145-153, 1999.

[45] Verma N.K., Chakraborty A. and Chatterjee A.K., "Joint replacement of multi retailer with variable replenishment cycle under VMI", European Journal of Operation Research, 233, pp. 287-289, 2014.

[46] Viswanathan S. and Piplani R., "Coordinating supply chain inventories through common replenishment epochs", European Journal of Operation Research, 129(2), pp. 277-286, 2001.

[47] Waller M., Johnson M.E. and Davis T., "Vendor-managed inventory in the retail supply chain", Journal of Business Logistics, 20, pp. 183-204, 1999.

[48] Williams M.K., "Making consignment-and vendor-managed inventory work for you", Hospital Material Management Quarterly, 21(4), pp. 59-63, 2000.

[49] Taleizadeh A.A., Niaki S.T.A., Shafi N., Ghavamizadeh-Meibodi R. and Jabbarzadeh A., "A particle swarm optimization approach for constraint joint single buyer-single vendor inventory problem with changeable lead-time and (r, Q) policy in supply chain", Int. J. Adv. Manuf. Tech., 51, pp. 1209-1223, 2010.