

A Comparative Study of Periodic-Review Order-Up-To (T, S) Policy and Continuous-Review (s, S) Policy in a Serial Supply Chain over a Finite Planning Horizon

P. V. Rajendra Sethupathi, Chandrasekharan Rajendran
and Hans Ziegler

Abstract In this paper, we consider a serial supply chain (SC) operating with deterministic and known customer demands and costs of review or orders, holding, and backlog at every installation over a finite planning horizon. We present an evaluation of two order policies: Periodic-review order-up-to S policy (i.e., (T, S) policy), and (s, S) policy. We first present a mathematical programming model to determine optimal re-order point and base-stock for every member in the SC. By virtue of the computational complexity associated with the mathematical model, we present genetic algorithms (GAs) to determine the order policy parameters, s and S for every stage. We compare the performances of GAs (for obtaining installation s and S) with the mathematical model for the periodic-review order-up-to (T, S) policy that obtains in its class optimal review periods and order-up-to levels. It is observed that the (s, S) policy emerges to be mostly better than the (T, S) policy.

Keywords Serial supply chain • Inventory management • Periodic-review order-up-to S (T, S) policy • (s, S) policy • Mathematical programming models • Genetic algorithms

P. V. R. Sethupathi (✉)
Vaigai College of Engineering, Therkuthuru Melur, Madurai 625122,
India
e-mail: pvrsiitm@yahoo.co.in

C. Rajendran
Department of Management Studies, Indian Institute of Technology Madras,
Chennai 600036, India
e-mail: craj@iitm.ac.in

H. Ziegler
Department of Production and Logistics, Faculty for Business Administration
and Economics, University of Passau, 94032 Passau, Germany
e-mail: ziegler@uni-passau.de

1 Introduction

Supply chain (SC) decisions are broadly classified into strategic, tactical, and operational decisions, and the operational perspective can be addressed in terms of four problem areas, namely, inventory management and control; production, planning, and scheduling; information sharing and coordination monitoring; and operation tools (Ganeshan et al. 1999). Inventory is held by installations or members in a SC in different forms so as to provide continuous service to the respective downstream customer and finally to customers/consumers. Love (1979) defined inventory as goods or materials in the control of an enterprise, and held for a time in a relatively idle or unproductive state, awaiting its intended use or sale. Efficient and effective management of inventory throughout the SC significantly improves the service provided to the customer (Lee and Billington 1992). Nahmias (2008) listed out the motivation for holding inventories in view of economies of scale, uncertainties, smoothing, transportation, control cost, and logistics. Some issues that are critical in the SC inventory management include the inventory order policy such as the periodic-review order-up-to policy (T, S) and continuous-review (s, S) policy. These order policies help answer basically two questions: When to order and how much to order? The objective is to minimize the sum of costs, mostly consisting of costs of review or order, holding, and shortage at different members in the SC. In this paper, a serial SC that manufactures a single product with discrete customer demands known a priori over the finite planning horizon is considered, and a relative evaluation of two inventory order policies, namely, periodic-review order-up-to policy (i.e., (T, S) policy) and the (s, S) policy, is presented by considering many SC settings. It is possibly for the first time in the literature that such a relative evaluation of such order policies is undertaken in a serial SC over a finite time horizon.

2 Literature Review

Clark and Scarf (1960) addressed the problem of determining optimal purchasing quantities in a serial multi-echelon system so as to minimize the long-run average cost, and showed that an echelon base-stock policy is optimal for the finite horizon problem. Federgruen and Zipkin (1984) extended Clark and Scarf's work to infinite horizon problem, and proved that a stationary order-up-to-level policy is optimal. Lee and Billington (1993) developed a decision support system for Hewlett-Packard company to aid inventory and service benchmarking, operational planning and control, and what-if analyses. Axsäter and Rosling (1993) compared the installation and echelon stock policy, and proved that when every stock-point in a multi-echelon inventory system is controlled by an order-up-to policy, an installation stock policy can always be replaced by an equivalent echelon stock policy, and vice versa. Glasserman and Tayur (1995) developed simulation-based

methods to estimate sensitivities of inventory costs with respect to policy parameters. Gallego and Zipkin (1999) discussed the issue of stock positioning, and constructed heuristics to minimize the system average cost. Zipkin (2000) presented discussions on base-stock levels for series systems, assembly systems, and distributed systems with local control and central control.

Min and Zhou (2002) classified SC models into inventory theoretic and simulation hybrids, and stressed the need of mathematical programming models to explore multi-echelon and multi period issues in the SC. Shang and Song (2003) developed an easily implementable heuristic to determine the echelon base-stock levels for an N -stage serial system by solving $2N$ single stage news vendor-type problems. Daniel and Rajendran (2005a) developed a simulation-based heuristic that attempts to optimize (in the class of base-stock policy) the base-stocks in a serial SC with the objective of minimizing the total SC cost consisting of holding costs and shortage costs at all installations in the SC. They proposed a genetic algorithm (GA) to determine the best installation base-stocks for a serial SC. Daniel and Rajendran (2005b) developed a problem-specific heuristic and a simulated annealing (SA) heuristic for finding the best installation base-stocks in a serial SC. Daniel and Rajendran (2006) developed different variants of GA to obtain the best installation base-stocks for a serial SC, and compared these variants with the complete enumeration and the best-move local search technique. However, they did not consider the presence of order costs in the SC.

Another application of GAs to SC inventory optimization was reported by Haq and Kannan (2006) by considering a two-echelon distribution-inventory SC for the bread industry. Cheng et al. (2006) proposed using a fuzzy inventory controller to determine the ordering quantity for the members in a SC. Axsäter (2006) provided discussions on multi-echelon inventory ordering policies including the determination of optimal lot sizing and re-order points. van Houtum et al. (2007) considered a single-item, periodic-review, serial inventory/production system, with linear inventory-holding and penalty costs, and proved the optimality of base-stock policies by deriving newsboy equations for the optimal base-stock levels, and described an efficient exact solution procedure for the case with mixed Erlang demands. Shang and Song (2007) considered two models of stochastic serial inventory systems and showed that the optimal policy parameters can be bounded and approximated by a series of independent, single-stage optimal policy parameters. Cheung and Zhang (2008) considered a SC with one supplier and multiple retailers in which base-stock policies are practiced, specifically two replenishment strategies: Synchronized ordering and balanced ordering. Johansen and Thorstenson (2008) extended the problem of determination of optimal base-stock of the inventory system with continuous review and constant lead time to the case with periodic review and stochastic and sequential lead times. For a detailed review of literature, see Sethupathi and Rajendran (2010).

Most studies considered only holding and shortage costs, and they did not consider the order costs to be significant which may not be true in all SCs; in that case, the total costs consist of holding, shortage, and order costs at various echelons in the SC. Sethupathi and Rajendran (2010) modeled a serial SC with

periodic-review order-up-to policy (i.e., (T, S) policy), and determined the optimal and heuristic review periods and base-stocks at installations in the SC (with deterministic customer demands over a fine planning horizon) respectively using a mathematical programming model and two GAs. It is evident that the determination of order-up-to levels in SCs and the minimization of the total SC cost are quite complex and computationally quite tedious (e.g., see Lee and Billington 1993; Petrovic et al. 1998; Shang and Song 2003; Daniel and Rajendran 2006; Sethupathi and Rajendran 2010); it is also evident from the literature review that the determination of optimal order quantity and re-order level or re-order point is computationally difficult with the consideration of the exact total costs in SC, and that no attempt has been made to determine such optimal order policy parameters in a SC in a deterministic demand environment over a finite planning horizon (also see the related observations of researchers in the case of determination of inventory order policy parameters with respect to periodic-review base-stock policy in a serial SC).

The present study aims to model a serial SC operating with dynamic, known, and deterministic customer demands over a finite planning horizon: First with the consideration of (s, S) policy with re-order point to determine the installation base-stock and re-order point for every member in the SC; and compare its performance with that of periodic review order-up-to S policy (i.e., (T, S) policy) that operates with base-stock and review period for every installation. We make use of mathematical programming models to determine the optimal parameter values in the respective class of inventory order policies, and subsequently three GAs to determine the heuristic parameter values (via deterministic simulation) in view of the computational complexity associated with the mathematical programming models. We consider the installation inventory order policy and its parameters in view of the ease of implementation in the proposed mathematical programming model and the GAs. Our model consists of a serial SC with a supplier, a manufacturer, a distributor, and a retailer, and a time unit is assumed to be discrete and it corresponds to a day, though not restrictive. The present work employs the SC model similar to the one by Sethupathi and Rajendran (2010); however differs in terms of the order policy i.e., (s, S) policy considered in this study.

3 Mathematical Programming Model for the (s, S) Policy

The SC framework considered in this study comprises four members, namely, retailer, distributor, manufacturer, and supplier, and the material flow is shown in Fig. 1. This study primarily focuses on the determination of the optimal or heuristic best installation base-stocks and re-order points, i.e., S and s respectively at every installation, and hence we do not treat the customer as a member in the context of the current problem, except that a customer demand triggers the information flow and hence the material flow in the SC. All four members add value to the product as it passes through the SC before it is delivered to the

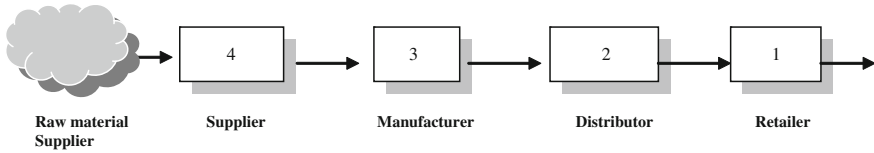


Fig. 1 Material flow in the serial supply chain

customer. The members are linked through information flow in both directions and products flow from the most upstream member to the lowest downstream member. The whole SC works with a pull strategy and the customer demand arising at member 1 (i.e., retailer) pulls the product from the subsequent upstream member. Inventory is controlled by the SC members by using the (s, S) policy with a re-order point. This review mechanism triggers replenishment orders at every member depending upon the inventory on hand, pre-specified installation base-stock, re-order point, outstanding orders, and backorders at that member. Every member continuously monitors its installation inventory position; when it equals or falls below the re-order point s , an order is triggered to the immediate upstream member. Different holding and shortage cost-rates and ordering costs exist at different SC members. Every member has a deterministic replenishment lead time equal to the sum of production lead time and transportation lead time with respect to the upstream member.

3.1 Model Assumptions

1. A single product flows through the SC.
2. Time is assumed to be discrete, and the unit of discrete time is assumed to be a day.
3. We consider a serial SC model comprising N installations with a finite time horizon and with known discrete customer demands varying over time.
4. All the N installations or stages or members in the SC operate under an installation (s, S) policy with the respective re-order points and base-stocks for every member in the SC.
5. Inventory order policy parameters such as the base-stock (S) and re-order point (s) for a given member remain the same across the entire finite time horizon.
6. Base-stock and re-order point for a given member or installation in the SC take integer values as the customer demands are assumed to be integers.
7. The retailer faces a customer demand, which is assumed to be stationary and uniformly distributed in the interval $[20, 60]$ in the computational experiments considered in this study (though this demand distribution is not restrictive). Customer demands are sampled from the given distribution, and they are assumed to be discrete and known over the finite planning horizon in this study.

8. Lead time for information or order processing is zero.
9. Processing (procurement/production/packing) lead time and transportation lead time are combined accordingly at each stage and considered together as one component, called replenishment lead time for that installation or member in the SC, and is assumed to be deterministic.
10. There is no lot-size or discount policy for members in the SC.
11. Every member in the SC has its own installation or local holding and shortage cost rates, and an order cost per order.
12. Re-order point of a member is assumed to be less than or equal to the base-stock of that member.
13. If the demand exceeds on-hand inventory at a member, then the excess demand is backlogged.
14. Availability of raw material for installation N , i.e., the supplier, is unlimited.
15. All installations have infinite capacity.
16. The entire customer demand on day t is assumed to occur during the time epoch Δt within day t itself; the demand from installation j gets transmitted to upstream member $(j + 1)$ immediately at the end of time epoch Δt , if the installation inventory position equals or falls below the re-order point s_j at installation j . Hence, relevant inventory information gets updated at the beginning of and at the end of t at every installation.

3.2 Supply Chain Model Description with (s, S) Policy

The sequence of events taking place at installation or member j , for $j = 1$ to N in the same order, in time (or during day) t , where N is the number of installations in the SC, is as follows.

- (1) The receipt of material at member j (shipped from member $(j + 1)$ to member j) takes place if the material is due to arrive at the beginning of current time instant t . The installation's inventory information, namely, on-hand inventory, is updated.
- (2) Member 1 receives the customer demand; for member j , $j > 1$, the order from the immediate downstream member is received, when the inventory position of installation $(j - 1)$ has reached or has fallen below its re-order level.
- (3) The possible order fulfillment (combining both installation's backorder as of the previous day and the current day's demand received from downstream member $(j - 1)$) takes place, depending upon the available on-hand inventory at stage j . If sufficient on-hand inventory is not there with member j to meet this combined demand, then the unsatisfied demand is backlogged. The downstream member $(j - 1)$ receives a partially or fully fulfilled order quantity after a delay corresponding to replenishment lead time of member $(j - 1)$. If the downstream member is the customer, then the customer receives it in the current day t itself. Installation inventory at stage j is updated.

- (4) With the available local inventory information, member j triggers an order to the upstream member $(j + 1)$ only if the inventory position of j has reached or fallen below the re-order level of j ; otherwise no order placement takes place. The order placement depends upon the inventory on hand, pre-specified base-stock S_j , re-order point s_j , outstanding order, and backorder at that member j . If order placement takes place, member $(j + 1)$ realizes the order placed by member j immediately, since the information/order processing lead time is assumed to be zero.
- (5) Installation inventory such as on-hand inventory, on-order inventory, and backorder are updated, and the sum of local holding cost, shortage cost, and order cost, if an order takes place, are computed with respect to member j .

Since it is assumed in this study that a day corresponds to a unit time, t is incremented and the same sequence of events gets repeated on the following day; the same sequence of events takes place at all members in the SC, for $j = 1, 2, \dots, N$. The total supply chain cost (TSCC) is computed over the given finite-time horizon of T days.

3.3 Formulation of the Mathematical Model for the (s, S) Policy

The notations used in the mathematical model are as follows:

TSCC	total supply chain cost
j	installation/stage index
N	number of installations in the SC
T	total number of days (planning horizon) over which TSCC is computed
t	current time or current day
S_j	installation base-stock at installation j
s_j	installation re-order point for installation j
h_j	installation holding cost-rate for installation j
b_j	installation shortage cost-rate for installation j
O_j	installation ordering cost for installation j
LT_j	installation replenishment lead time with respect to installation j /* note that the member j receives on t , the possible shipment from $(j + 1)$ that has taken place at the end of $(t - LT_j)$; hence in this study, when we assume $LT_j = 1$, it means that member j receives on day t the shipment that has been shipped from member $(j + 1)$ on day $(t - 1)$, though theoretically LT_j equals 0; however, we set $LT_j = 1$ and we carry on with this setting of LT_j for the sake of correctness of mathematical formulation in this study; also see Daniel and Rajendran (2005a, b, 2006) */
$I_{j,t}$	end installation on-hand inventory at installation j at t

$I_{j,t}^*$	installation on-hand inventory at installation j at the beginning of t , after the possible replenishment from installation $(j + 1)$ arrives at installation j
$OI_{j,t}$	end on-order inventory at installation j at t
$OI_{j,t}^*$	on-order inventory at installation j at the beginning of t
$SUMDEM_{j,t}$	sum of demand at (i.e., received by) installation j up to t
$SUMORD_{j,t}$	sum of orders placed by installation j up to t
$B_{j,t}$	backorder at installation j at the end of t
$D_{j,j+1,t}$	demand placed by installation j to upstream installation $(j + 1)$ at t /* this demand is assumed to be immediately realized by installation $(j + 1)$ at t itself; $D_{0,1,t}$ corresponds to customer demand at installation 1 on day t ; as for other installations, $D_{j,j+1,t}$ equals zero if $(I_{j,t} + OI_{j,t}^* - B_{j,t} > s_j)$; otherwise it equals $(S_j - (I_{j,t} + OI_{j,t}^* - B_{j,t}))$ that is same as the sum of demands received by the installation up to t minus the sum its order quantities up to $(t - 1)$. */
$\delta_{j,t}^*$	a binary variable that assumes the value of 1, whenever there is an order placement at installation j at t ; else it is 0
$QS_{j,j-1,t}$	quantity shipped from installation j to $(j - 1)$ at the end of t
$QS_{N+1,N,t}$	raw material shipped to installation N at the end of t /* installation $N + 1$ is assumed to have raw material supply of infinite capacity */

The objective is to minimize the total system-wide cost given as follows:

$$\text{Minimize TSCC} = \sum_{t=1}^T \sum_{j=1}^N \left(h_j I_{j,t} + b_j B_{j,t} + O_j \delta_{j,t}^* \right) \quad (1)$$

subject to the following:

{
{

$$OI_{j,t}^* = OI_{j,t-1} - QS_{j+1,j,t-LT_j} \quad (2)$$

$$I_{j,t}^* = I_{j,t-1} + QS_{j+1,j,t-LT_j} \quad (3)$$

$$I_{j,t} - B_{j,t} = I_{j,t}^* - B_{j,t-1} - D_{j-1,j,t} \quad (4)$$

$$SUMDEM_{j,t} = SUMDEM_{j,t-1} + D_{j-1,j,t} \quad (5)$$

$$OI_{j,t}^* + I_{j,t} - B_{j,t} \leq s_j + M(1 - \delta_{j,t}^*) \quad (6)$$

$$OI_{j,t}^* + I_{j,t} - B_{j,t} \geq s_j + 1 - M\delta_{j,t}^* \quad (7)$$

$$D_{j,j+1,t} \leq SUMDEM_{j,t} - SUMORD_{j,t-1} + M(1 - \delta_{j,t}^*) \quad (8)$$

$$D_{j,j+1,t} \geq \text{SUMDEM}_{j,t} - \text{SUMORD}_{j,t-1} - M(1 - \delta_{j,t}^*) \quad (9)$$

$$D_{j,j+1,t} \leq M\delta_{j,t}^* \quad (10)$$

$$\text{QS}_{j,j-1,t} = I_{j,t}^* - I_{j,t} \quad (11)$$

$$\text{SUMORD}_{j,t} = \text{SUMORD}_{j,t-1} + D_{j,j+1,t} \quad (12)$$

$$\text{OI}_{j,t} = \text{OI}_{j,t}^* + D_{j,j+1,t} \quad (13)$$

}, for $j = 1, 2, \dots, N$

$$\text{QS}_{N+1,N,t} = D_{N,N+1,t} \quad (14)$$

}, for $t = 1, 2, \dots, T$

with initial conditions as

$$I_{j,0} = S_j, \quad \forall j \leq N \quad (15)$$

$$s_j \leq S_j, \quad \text{for } \forall j \leq N \quad (16)$$

$$\text{OI}_{j,0} = 0, \quad \forall j \leq N \quad (17)$$

$$B_{j,0} = 0, \quad \forall j \leq N \quad (18)$$

$$\text{SUMDEM}_{j,0} = 0, \quad \forall j \leq N \quad (19)$$

$$\text{SUMORD}_{j,0} = 0, \quad \forall j \leq N \quad (20)$$

$$\text{QS}_{j+1,j,t-LT_j} = 0, \quad \forall t \leq LT_j \text{ and } \forall j \leq N \quad (21)$$

$$\delta_{j,t}^* \in \{0, 1\} \forall j \leq N \text{ and } \forall t \leq T \text{ and all other variables } \geq 0, \forall j \leq N \text{ and } \forall t \leq T. \quad (22)$$

Equation 1 shows the objective function to minimize the TSCC comprising the holding, shortage, and ordering costs for all installation over T days. Equation 2 updates the on-order inventory at j at the beginning of t . Equation 3 updates the on-hand inventory at j at the beginning of t . Equation 4 updates the on-hand inventory/backorder at j after the demand realization. This equation holds, in view of our assumption that $b_j \geq h_{j+1}$. However, if this assumption does not hold, both terms may co-exist in Eq. 4, and hence we need to have the following expressions introduced to avoid the co-existence of $I_{j,t}$ and $B_{j,t}$:

$$I_{j,t} \leq M\delta_{j,t}^{**} \text{ and } B_{j,t} \leq M(1 - \delta_{j,t}^{**}), \text{ with } \delta_{j,t}^* \in \{0, 1\}, \text{ for } \forall j \leq N \text{ and } \forall t \leq T \quad (23)$$

where M denotes a large positive integer value.

Equation 5 updates the sum of demand up to t . Expressions 6 and 7 monitor inventory position at stage j (i.e., $\text{OI}_{j,t}^* + I_{j,t} - B_{j,t}$) and trigger the order if the

end-inventory position equals or falls below the re-order level at j (and in this case $\delta_{j,t}^*$ becomes equal to one and hence $D_{j,j+1,t} = \text{SUMDEM}_{j,t} - \text{SUMORD}_{j,t-1}$, i.e., $(S_j - (I_{j,t} + \text{OI}_{j,t}^* - B_{j,t}))$); else $\delta_{j,t}^*$ becomes zero and hence $D_{j,j+1,t} = 0$, indicating no order placement. Expressions 8, 9, and 10 set the order to the upstream member accordingly (note that the order quantity equals zero if no order placement is there). Note that since we assume demands to be integers, we have ‘1’ in Expression 7, and all variables in an optimal solution are also integers. Equation 11 shows the shipment quantity from member j to the downstream member. Equation 12 updates the sum of orders placed (or on-order inventory) at stage j up to t . Equation 13 computes the end on-order inventory at j at t . Equation 14 shows the immediate shipment of quantity from the raw material supplier (who has infinite raw material availability) to the supplier, namely, installation N . Equations 15–21 are constraints which are used for initial conditions. Equation 22 refers to the binary variable used to represent the order placement and non-negativity constraints.

3.4 Supply Chain Settings

Tables 1 and 2 show the experimental settings of the SC test problems with lead time settings and cost settings considered in this study.

We sample daily integer customer demands that are uniformly distributed between [20, 60], and treat them as known and deterministic over the finite planning horizon. We establish two different demand streams or patterns. One long stream of demands spanning over 1200 days (called FD in this study) is directly generated from a seed value through a uniform random number generator for sampling customer demands. Another long stream spanning over 1200 days (called AD in this study) of customer demands is generated by the antithetic uniform random number from the same seed value. For example, if one demand in the first run is 30, and then in the antithetic run demand equals 50 (in view of the demand distribution being U [20, 60]). The method of antithetic sampling is a commonly used procedure for negatively correlated pair of samples (Deo 1999). These demand streams are long and sufficient for substantive experiments. We assume three different deterministic lead time settings as given in Table 1. The different cost settings are assumed as follows:

Table 1 Lead time settings

Lead time setting	Lead time (in days)			
	Supplier (LT ₄)	Manufacturer (LT ₃)	Distributor (LT ₂)	Retailer (LT ₁)
LT1	1	1	1	1
LT2	2	2	2	2
LT3	3	3	3	3

Installation shortage cost rate, $b_j = C \times h_j$, where $C = 2, 4, 8, 16$; and installation ordering cost $O_j = K \times E(D) \times h_j$,

where $E(D)$ is the expected customer demand with respect to the customer demand distribution and $K = 2, 4, 8, 16$.

The rationale for this setting of O_j is that we relate the ordering cost to the setting of holding cost at stage j , and that we take the holding cost as the basis for the ordering cost because normally we deal with a SC with a very high service level (see Silver et al. 1998). See Table 2 for details of different cost settings where S, M, D, and R represent supplier, manufacturer, distributor, and retailer, respectively.

In all, we create 96 SC test problem instances in the present study (16 cost settings \times 3 lead time settings \times 2 demand patterns). It is to be noted that we increase the ratio of b_j/h_j with the installation order costs remaining the same (e.g., see CS1–CS4), and that we increase installation order costs with the ratio of b_j/h_j remaining the same (e.g., see CS1, CS5, CS9, and CS13). This is done in order to discover a possible pattern in the behavior of order policies (in terms of their base-stocks and re-order points) as a function of order costs and ratios of b_j/h_j .

Table 2 Supply chain settings with respect to supply chain members

Supply chain setting	Holding cost rate (h_j)				Shortage cost rate (b_j)				Ordering cost (O_j)				(C, K)
	S	M	D	R	S	M	D	R	S	M	D	R	
CS1	1	2	4	8	2	4	8	16	80	160	320	640	(2,2)
CS2	1	2	4	8	4	8	16	32	80	160	320	640	(4,2)
CS3	1	2	4	8	8	16	32	64	80	160	320	640	(8,2)
CS4	1	2	4	8	16	32	64	128	80	160	320	640	(16,2)
CS5	1	2	4	8	2	4	8	16	160	320	640	1280	(2,4)
CS6	1	2	4	8	4	8	16	32	160	320	640	1280	(4,4)
CS7	1	2	4	8	8	16	32	64	160	320	640	1280	(8,4)
CS8	1	2	4	8	16	32	64	128	160	320	640	1280	(16,4)
CS9	1	2	4	8	2	4	8	16	320	640	1280	2560	(2,8)
CS10	1	2	4	8	4	8	16	32	320	640	1280	2560	(4,8)
CS11	1	2	4	8	8	16	32	64	320	640	1280	2560	(8,8)
CS12	1	2	4	8	16	32	64	128	320	640	1280	2560	(16,8)
CS13	1	2	4	8	2	4	8	16	640	1280	2560	5120	(2,16)
CS14	1	2	4	8	4	8	16	32	640	1280	2560	5120	(4,16)
CS15	1	2	4	8	8	16	32	64	640	1280	2560	5120	(8,16)
CS16	1	2	4	8	16	32	64	128	640	1280	2560	5120	(16,16)

3.5 Results of Execution of Mathematical Model Through an Optimization Solver and the Need for Heuristic Algorithm

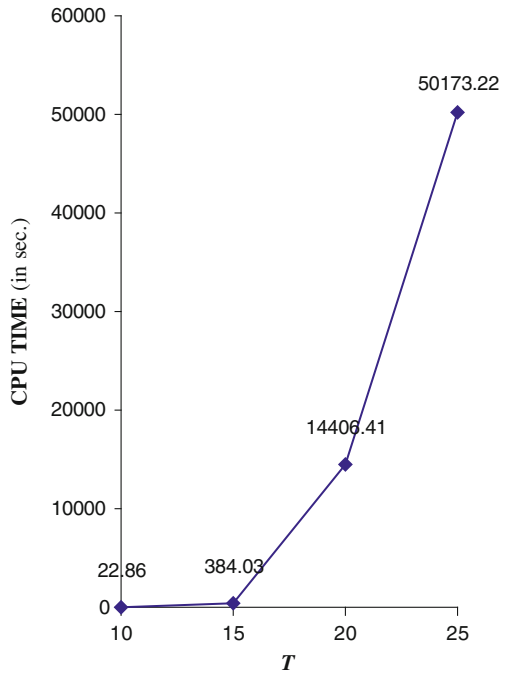
The proposed model considering problem instances with $T = 10$, $T = 15$, $T = 20$, $T = 25$, and $T = 30$ and with the consideration of CS1 and LT1 has been executed with CPLEX, an optimization solver, using a computer with an Intel Pentium IV processor of 3.0 GHz speed and 1 GB RAM. The solver is able to solve the problems with $T = 10$, $T = 15$, $T = 20$, and $T = 25$. When we set $T = 30$, the solver could obtain only an upper bound and not the optimal solution, and the solver could not solve this test problem even after 6 h of execution, and its execution got terminated due to its memory limitations. The base-stocks and re-order points, which lead to minimum SC costs, and the CPU times for the five test problems, are reported in Table 3, where S, M, D, and R represent supplier, manufacturer, distributor, and retailer, respectively. We have also obtained the LP-relaxed solution, i.e., a lower bound on TSCC, by relaxing the binary constraints in the mathematical programming model (i.e., by treating $\delta_{j,t}^*$'s as continuous variables in the interval (0,1)). The lower bounds thus obtained for the test problems are also reported in Table 3. As the lower bounds obtained in these problems appear to be weak, we do not consider the lower bound through the LP-relaxation in our further analyses.

Figure 2 shows the computational effort of the solver for test problems with $T = 10, 15, 20$, and 25 . From Fig. 2, it can be noted that the computational time taken for obtaining solutions appears to increase exponentially. This is not unexpected due to the presence of binary variables in the mathematical programming model. Wang (2011) also presented a similar observation that when time is discrete, the large dimension of the inventory state for the exact formulation usually precludes an exact solution and hence the focus is shifted to approximate solutions. Hence, we resort to the use of a heuristic algorithm in the present study to obtain the best inventory parameters in the SC.

Table 3 Results obtained for the test problems using the solver

T	Base-stock levels				Re-order levels				TSCC	Solution status	CPU time (in s)	LP-relaxation-based lower bound
	S	M	D	R	S	M	D	R				
10	195	108	86	85	0	0	0	0	9454	Optimal	22.86	473.84
15	78	29	78	83	0	0	0	6	14148	Optimal	384.03	687.44
20	63	44	57	93	0	0	0	8	19199	Optimal	14406.41	873.44
25	49	50	57	93	0	0	0	8	23564	Optimal	50173.22	1100.24
30	799	61	320	98	0	0	0	13	49864	Upper bound	19642.69	1392.80

Fig. 2 Computational experience with test problems (setting CS1 + LT1)



4 Proposed Genetic Algorithms for the (s, S) Policy

GAs are search algorithms based on the mechanism of natural selection and natural genetics. Simplicity of operation and power of effect are two of the main attractions of the GA approach (Goldberg 1989). GA is applied as an optimization tool to a variety of SC problems. Attempts had been made to develop meta-heuristics such as GA for SC inventory optimization. Daniel and Rajendran (2006) developed simulation-based heuristics to obtain inventory levels (base-stock levels) in a serial SC with the objective of minimizing the total SC cost in the class of base-stock policy. They proposed GAs to determine the best base-stock levels for a serial SC, and their findings indicated the performance of GAs based on gene-wise crossover operators superior to the SA algorithm by Daniel and Rajendran (2005b). They found their GGA (gene-wise GA) to perform very well. Haq and Kannan (2006) and Kumanan et al. (2007) employed GAs for minimizing the total cost consisting of production, inventory, and distribution costs. Some applications of GAs in SCs/multi-echelon inventory systems were discussed by Berry et al. (1998), Zhou et al. (2002), Lee et al.(2002), Hong and Kim (2009), and Rom and Slotnick (2009); and Sethupathi and Rajendran (2010) considering (T, S) policy. For all these reasons, we have gone for GAs to obtain the best heuristic order policy parameters in our study in view of the well-established performance of GAs in supply-chain operations optimization. In this

study, we present the modified gene-wise GA (called MGGA), and two hybrid GAs, called HGA1 and HGA2 (GAs with the hybridization of gene-wise crossover operator and arithmetic crossover operator), to determine the best installation base-stocks and re-order points in the present case of (s, S) policy. It is to be noted that our present study is different from earlier works such as those by Daniel and Rajendran (2005a, 2006) in that earlier researchers did not consider the presence of order costs and hence they did not investigate the performance of policies such as (s, S) policy.

4.1 Notations and Terminologies for the Proposed GAs

The notations and terminologies used in the proposed GAs are as follows:

no_gen	number of generations
l	length of a chromosome (or the number of genes)
pop_size	population size
n	number of chromosomes (equal to pop_size)
par_pop	parent population, consisting of n parent chromosomes
S_j	installation base-stock at member j , where $j = 1$ to N
s_j	installation re-order point at installation j /* it is ensured in the implementation of all GAs that $s_j \leq S_j$ */
S_j^{UL}	upper limit on the base-stock at installation j /*(set rather loose to 1,000 in this study)*/
S_j^{LL}	lower limit on the base-stock at installation j /*(set to 20 in view of demand distribution assumed in the computational experiments in this study)*/
s_j^{UL}	upper limit on the re-order point /*(set to S_j in this study)*/
s_j^{LL}	lower limit on re-order point /*(set to 0 in this study)*/
f_k	fitness value for the k th chromosome
P_k	probability of selecting chromosome k into the mating pool (relative fitness of the k th chromosome)
u	a uniform random number between 0 and 1
CR	probability of crossover or Crossover Rate (CR) /*(set to 1 in our study)*/
MR	probability of mutation or Mutation Rate (MR)
R_m	merging rate used in HGA2
P_m	merging probability used in HGA2
int_pop	intermediate population consisting of child chromosomes or offspring that are obtained from the crossover of parent chromosomes in par_pop
res_pop	resultant population consisting of offspring after mutation
S_j^{old}	base-stock at installation j before mutation

- S_j^{new} base-stock at installation j after mutation
- s_j^{old} re-order point at installation j before mutation
- s_j^{new} re-order point at installation j after mutation

4.2 Mechanism of the Proposed GAs

The mechanism of MGGA, HGA1, and HGA2 is same except they vary in their respective crossover operators. All other steps are common for these three GAs.

4.2.1 Representation of a Chromosome

GA is a population-based search technique that works on a population represented by several individual chromosomes (solutions). Figure 3 shows the representation of a chromosome in our study as an example. It has eight genes of which the first four genes represent base-stocks, and the next four genes represent the corresponding re-order points of the supplier, manufacturer, distributor, and retailer, respectively. Genes 1 and 5 correspond to the supplier (representing respectively its base-stock and re-order point); genes 2 and 6 correspond to the manufacturer (representing respectively base-stock and re-order point), and so on.

4.2.2 Initialization of Population

The initial population *par_pop* is created by generating solutions randomly within limits with respect to base-stock and re-order point of the corresponding member. The lower limit on base-stock S_j^{LL} for every member j is set as their minimum

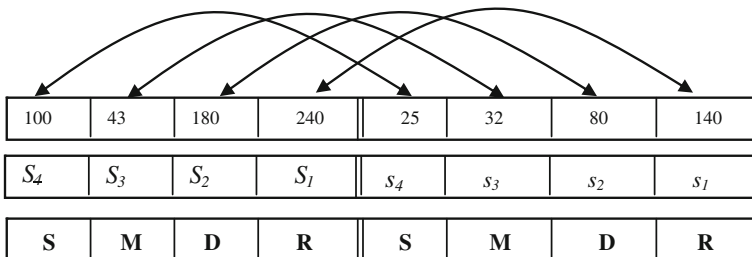


Fig. 3 Representation of the chromosome

S_4	S_3	S_2	S_1	s_4	s_3	s_2	s_1
200	95	120	245	120	40	80	125
:	:	:	:	:	:	:	:
142	66	180	125	25	15	100	65
:	:	:	:	:	:	:	:
28	48	140	320	10	22	75	180
:	:	:	:	:	:	:	:
190	85	108	127	85	24	66	19

Fig. 4 Initialization of population

customer demand which is equal to 20 in our computational experiments. The upper limit on base-stock S_j^{UL} for every member j is set as 1,000. We have fixed such a loose upper limit for base-stocks in order to test the robust performance of GAs in the large search space across all settings. The upper limit on the re-order point s_j^{UL} is set to S_j and the lower limit on re-order point s_j^{LL} is set to 0 in this study. The number of chromosomes (n) generated in the initial population, pop_size , is fixed as 40 (five times the length of the chromosome). For all n chromosomes, base-stocks and re-order points are generated randomly between their respective lower and upper limits. Figure 4 shows one such generated initial population.

4.2.3 Evaluation and Selection of Chromosomes for Crossover

Chromosomes in par_pop are evaluated through the deterministic simulation of SC using the demands sampled from the uniform distribution [20, 60] and known apriori over T days and their respective objective function values in terms of TSCC are obtained. For the sake of generality, let us use $TSCC_k$ corresponding to chromosome k . Fitness value f_k is computed for the k th chromosome by making use of the objective function value, i.e., set $f_k = 1/(1 + TSCC_k)$. Based on f_k values, the chromosomes are selected probabilistically for placement in the mating pool for crossover operation. The selection of chromosomes for the mating pool is done by using the roulette-wheel procedure (Goldberg 1989). The probability of selecting the k th chromosome from par_pop to the mating pool, P_k (i.e., relative fitness), is obtained by computing $(f_k / \sum_{k'=1}^n f_{k'})$. Similarly, P_k 's are calculated for all n chromosomes in par_pop and their respective cumulative probabilities are obtained.

4.2.4 Crossover Operators

We present three crossover operators in our GAs that employ arithmetic and gene-wise crossover operators either separately or in combination. The crossover rate CR is set to one in this study because the crossover operators are population-based crossover operators and we generate as many offspring as the number of parent chromosomes.

Modified Gene-wise Genetic Algorithm (MGGA)

The MGGA makes use of the crossover operator, called gene-wise crossover operator (see Daniel and Rajendran (2006)) that makes use of the information of genes of all the chromosomes that are present in par_pop , and builds an offspring out of those chromosomes. In their study, Daniel and Rajendran presented the superiority of the GGA to other crossover operators. We have chosen to adapt their GGA in our work (called the MGGA in our study) and the modified gene-wise crossover operator is explained with the example in Fig. 5. Assume that five chromosomes C1 to C5 are present in par_pop for the sake of illustration.

The gene-wise crossover operator constructs a child chromosome or offspring by considering the genes of all the chromosomes on the basis of its respective fitness function values. Assume that the probabilities of selection (or relative fitness values) of chromosomes (P_k) are 0.30, 0.25, 0.20, 0.15, and 0.10, respectively. Four uniform random numbers are generated because we need to construct one offspring with the first four genes representing the base-stocks and the next four corresponding genes

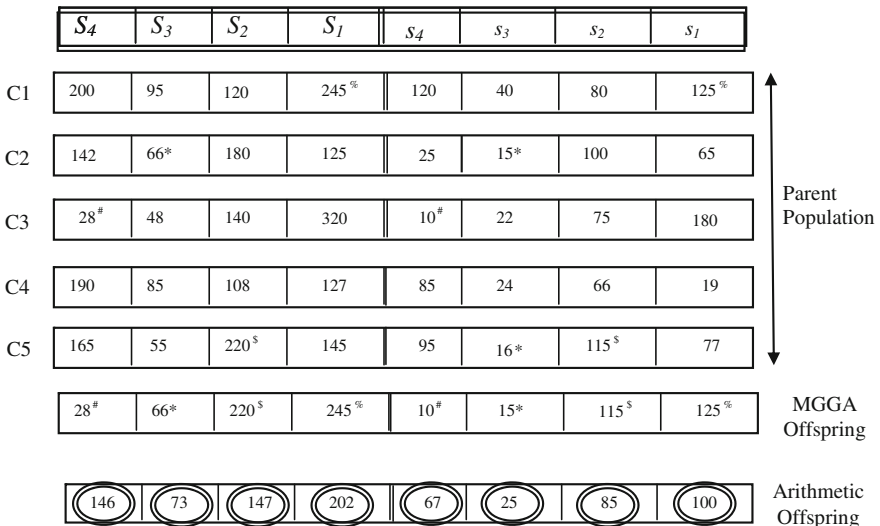


Fig. 5 An example of MGGA crossover operator and arithmetic operator with their parent population and the resultant offspring

representing re-order points. Let the random numbers be 0.61, 0.42, 0.93, and 0.14. Four parent chromosomes corresponding to these random numbers are selected as follows. The chromosome corresponding to $u = 0.61$ is chromosome C3 as seen from the cumulative probabilities of choosing chromosomes. Similarly, the chromosomes selected corresponding to the other three random numbers are chromosomes C2, C5, and C1, respectively. The offspring is now constructed gene-by-gene, by using the selected above four chromosomes in the same order. The first and fifth genes for the offspring are picked from the first selected chromosome C3, namely, {28, 10}. To construct the offspring's second and sixth genes, the second and sixth gene positions of the second selected chromosome C2 are chosen, i.e., {66, 15}. In this way we have the resultant offspring {28, 66, 220, 245, 10, 15, 115, 125}. Thus the generated offspring inherits eight genes from the four selected chromosomes in *par_pop* based on their fitness values. As the base-stock and re-order point of each member have inter-relationship ($s_j \leq S_j$), we inherit both base-stock and re-order point of a given member from the same parent into the offspring. In this way, the GGA proposed by Daniel and Rajendran (2006) is modified in our study taking into account both base-stocks and re-order points of members. The proposed gene-wise crossover operator may produce a good-quality offspring as the gene-wise crossover operator builds an offspring gene-by-gene, by making use of the fitness function values of all chromosomes in the parent population. This feature of the crossover operator leads to the generation of an offspring with proper and logical inheritance of both base-stock and re-order point from a given parent chromosome. This process is repeated until n offspring are generated.

Hybrid Genetic Algorithm 1 (HGA1)

Hybrid Genetic Algorithm 1 (HGA1) is an adapted version of the HGA proposed by Sethupathi and Rajendran (2010). In their work, Sethupathi and Rajendran employed a crossover operator using a combination of the arithmetic crossover operator and the gene-wise crossover operator to obtain the best heuristic order-up-to levels and review periods at installations. In the present work, we adapt their HGA to determine the best base-stocks and re-order points at installations. A chromosome is first obtained from the entire population by using the arithmetic crossover operator as follows. Assume that five chromosomes C1 to C5 are present in *par_pop* as shown in Fig. 5. Let the relative fitness values of these chromosomes be 0.30, 0.25, 0.20, 0.15, and 0.10. We make use of these values to construct the arithmetic offspring. The first gene in the offspring is constructed by an arithmetic operation (i.e., the first gene value of the first chromosome multiplied by its relative fitness plus the first gene value of the second chromosome multiplied by its relative fitness, and so on to obtain the first gene value of the offspring). For example, the value of first gene in the offspring is as follows:

$$((200 \times 0.3) + (142 \times 0.25) + (28 \times 0.2) + (190 \times 0.15) + (165 \times 0.1)) = 146$$

Likewise, every gene in the offspring is obtained by using this arithmetic operation on respective gene values of the respective chromosomes in *par_pop*. This offspring is placed first in the intermediate population *int_pop*, and this arithmetic chromosome is shown in Fig. 5. All the other offspring are constructed by the combination of gene-wise crossover operator and arithmetic crossover operator, as shown with examples in Fig. 6. For this, we proceed as follows: First, as per the procedure of the crossover operator in the MGGA, a chromosome is constructed by using the gene-wise crossover operator; then by selecting a set of positions, the respective arithmetic gene value gets superimposed on the gene value obtained by gene-wise crossover operator. By this procedure, we can obtain different offspring by filling up a set of genes with arithmetic crossover operator and the remaining genes by gene-wise crossover operator. For example, we can fill up one couple of positions by the arithmetic crossover operator and the remaining by the gene-wise crossover operator, and we can thus construct a total of 15 offspring in *int_pop* by the combination of both crossover operators. The remaining 25 offspring are built by using the gene-wise crossover operator presented in the MGGA. Figure 6 shows examples of thus constructed offspring by using the crossover operations in HGA1.

Hybrid Genetic Algorithm 2 (HGA2)

Hybrid Genetic Algorithm 2 (HGA2) employs a crossover operator which uses a combination of the arithmetic crossover operator and the gene-wise crossover operator. As in HGA1, a unique chromosome is first obtained from the entire population by using the arithmetic crossover operator (see Fig. 5). After the construction of a chromosome using the arithmetic crossover operator, the remaining chromosomes in *int_pop* are constructed as follows: First, a chromosome is constructed by using the gene-wise crossover operator as done in MGGA; then, every gene in this constructed chromosome is subjected to a probability of merging P_m and merging rate R_m (set as 0.25 and 0.3 respectively after a pilot study). Considering the first four genes corresponding to order quantities, one by one, a uniform random number u is generated; if u is $\leq P_m$, then the corresponding gene value will be altered as follows:

value of new gene =

$$\left(\begin{array}{l} (R_m \times \text{value of the gene obtained by arithmetic crossover operator}) \\ + (1 - R_m) \times (\text{value of the gene obtained by gene-wise crossover operator}) \end{array} \right)$$

otherwise, the old gene value constructed by the gene-wise crossover operator remains. Note that the value of the re-order point is also altered if the corresponding installation's base-stock is altered. For example, let the chromosomes 3, 2, 5, and 1 be randomly selected in the gene-wise crossover operation, and four uniform random

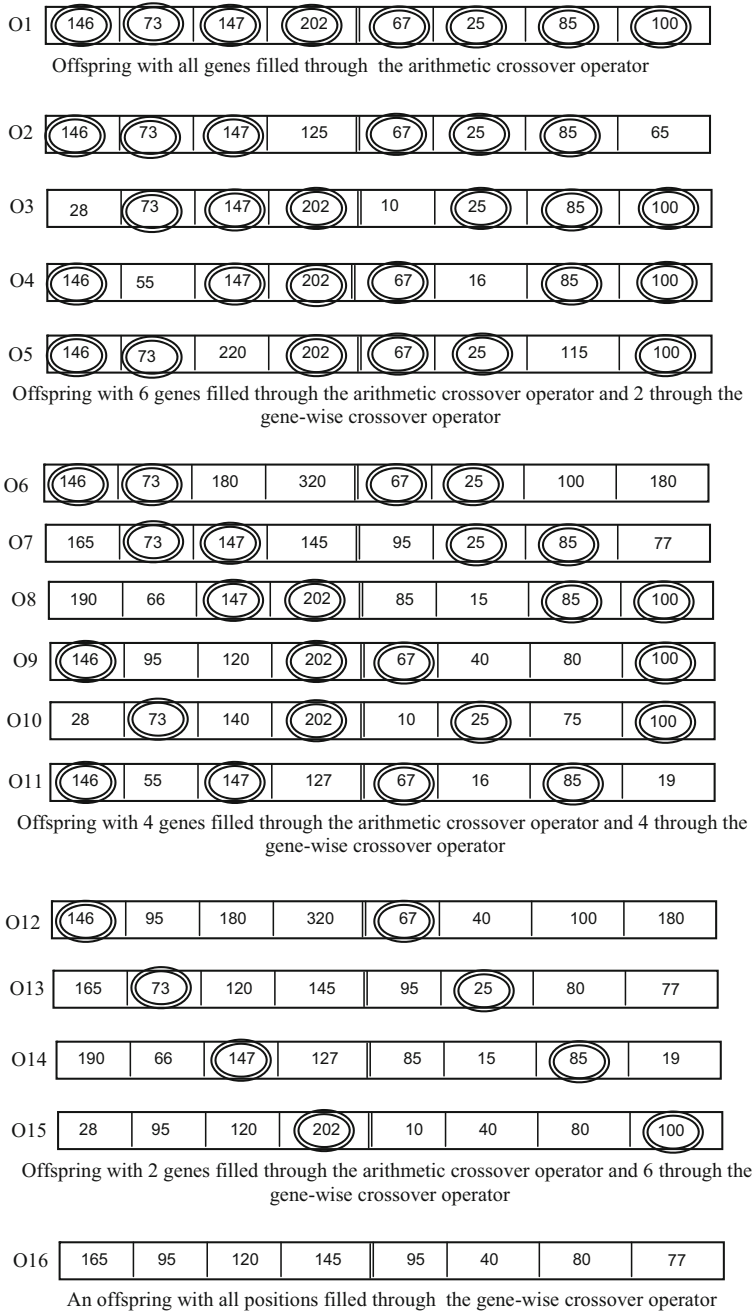


Fig. 6 Examples of the constructed offspring by the hybrid genetic algorithm 1 (HGA1)

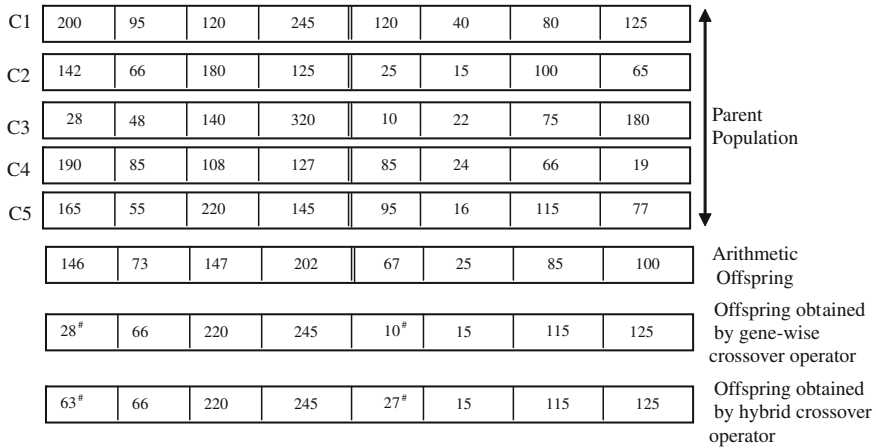


Fig. 7 An example of constructed offspring by the proposed hybrid genetic algorithm 2 (HGA2)

numbers sampled be 0.21, 0.38, 0.62, and 0.53. Gene positions 1 and 5 are altered, because the corresponding random number generated is less than or equal to P_m (i.e., 0.25). So the value of the first gene becomes $((0.3 \times 146) + (0.7 \times 28) = 63)$ and 5th gene becomes $((0.3 \times 67) + (0.7 \times 10) = 27)$, and the remaining genes are retained as constructed by the gene-wise crossover operator. The rationale of this approach of constructing offspring is to exploit the goodness of both arithmetic and gene-wise crossover operators. Figure 7 shows an example of the mechanism of the proposed hybrid crossover operator in HGA2. This process is repeated until all the offspring are built for *int_pop*.

4.2.5 Mutation Operation

The mutation operator used in this study is a gene-wise mutation. As the chromosomes are represented in a phenotypic manner, the first four genes are subjected to mutation with a probability of MR. When a gene is subjected to mutation, the value of the gene is altered as follows:

$$S_j^{new} = \left(\min \left\{ \max \left\{ (S_j^{old} \times (1 - x)) + (S_j^{old} \times 2 \times x \times u); S_j^{LL} \right\}; S_j^{UL} \right\} \right) \quad (24)$$

where u is a sampled uniform random number and $0 < x < 1$. We set $x = 0.2$ (after a pilot study). At the same time, we also mutate the corresponding gene containing the re-order point for that member which is altered as follows using the same u :

$$s_j^{new} = \left(\min \left\{ \max \left\{ (s_j^{old} \times (1 - x)) + (s_j^{old} \times 2 \times x \times u); s_j^{LL} \right\}; s_j^{new} \right\} \right) \quad (25)$$

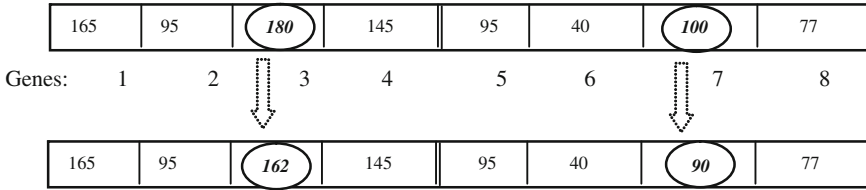


Fig. 8 Mutation operation

For example, consider the offspring shown in Fig. 8; let four random numbers be generated corresponding to the first four genes, and let them be 0.55, 0.46, 0.12, and 0.79, respectively. When the mutation rate (MR) is 0.2, the genes selected for mutation are the third gene and the seventh gene corresponding to this offspring. Again, a random number u is generated; let the value be 0.25. By using Equations 24 and 25, the values of third and seventh gene become 162 and 90, respectively. Here we use the same u that is sampled for mutating both base-stock and re-order point of the chosen installation so as to maintain proportionate perturbations in the base-stock and the corresponding re-order point. Figure 8 shows an example of how mutation is being carried out. As a result of mutation operation on the offspring in *int_pop*, we obtain the resultant pool called *res_pop*.

4.2.6 Survival of Chromosomes into the Next Generation

Now that there are n chromosomes in *par_pop* and n chromosomes in *res_pop*, we choose the best n distinct chromosomes out of these $2n$ chromosomes (present in *par_pop* and *res_pop*), on the basis of their TSCC values. The selected n distinct chromosomes become the parent chromosomes for the next generation. At the end of every generation, the best chromosome with the best TSCC value is subjected to a local search technique and updated.

4.2.7 Local Search Technique

We propose a local search on the best chromosome to enhance the convergence process. The best chromosome obtained at the end of Sect. 4.2.6 is removed from the population set and subjected to the following local search technique. We alter only one position randomly and with that generated neighborhood solution evaluate this setting of base-stocks and re-order points in the SC over the finite time horizon to find whether this neighborhood solution is better or worse than or same as that the seed solution. If the solution is better, the neighborhood solution replaces the seed solution; otherwise the seed chromosome is retained. For this purpose, a uniform random number u is generated between 0 and 1, and the couple of genes to be altered is decided as follows:

if $0 \leq u \leq 0.25$, then the gene-couple to be altered is 1 and 5;
 if $0.25 < u \leq 0.50$, then the gene-couple to be altered is 2 and 6;
 if $0.50 < u \leq 0.75$, then the gene-couple to be altered is 3 and 7; and
 if $0.75 < u \leq 1$, then the gene-couple to be altered is 4 and 8.

Again, a uniform random number u is generated between 0 and 1, and by using Equations 24 and 25 (similar to the mutation process), the selected gene positions are altered. Then the resultant chromosome is evaluated. If its TSCC value has a better value than seed chromosome's TSCC, then the seed chromosome is replaced by the generated chromosome; otherwise the same chromosome is retained. This procedure is repeated eight times (corresponding to the length of chromosome). The final chromosome thus obtained is placed in *par_pop* for the next generation.

4.2.8 Termination

The termination criterion is fixed in terms of number of generations. After 500 generations, the algorithm is terminated and the best chromosome at the end of termination provides the best base-stocks and re-order points for respective members in the SC which lead to the minimum TSCC of the solutions generated.

4.3 Step-By-Step Procedure of the GAs for the (s, S) Policy

- Step 1 Initialize $no_gen = 0$.
- Step 2 Generate the initial population with the number of chromosomes (n) equal to pop_size ($5 \times l$), and each chromosome representing base-stocks and re-order points at N installations in the SC.
- Step 3 Evaluate every chromosome in *par_pop* by evaluating them over the given finite time horizon in the SC, and hence obtain the objective function TSCC.
- Step 4 Obtain the fitness value f_k for every chromosome k , selection probabilities P_k 's and cumulative probabilities.
- Step 5 Do crossover operator as follows to obtain *int_pop*:
 - (a) In the MGGA, generate n offspring, by constructing each offspring directly from the chromosomes in *par_pop* by using the modified gene-wise crossover operator.
 - (b) In HGA1, construct one chromosome by the arithmetic crossover operator; generate $(n - 1)$ offspring by constructing each offspring directly from the chromosomes in *par_pop* by using the combination of both gene-wise crossover operator and arithmetic crossover operator or using gene-wise crossover, as appropriate.

- (c) In HGA2, construct one chromosome by the arithmetic operator; generate $(n - 1)$ offspring, by constructing each offspring directly from the chromosomes in par_pop , with the possible combination of both gene-wise crossover operator and arithmetic crossover operator and by using the probability of merging P_m and merging rate R_m .
- Step 6 Subject n chromosomes in int_pop to gene-wise mutation, with a probability of MR.
- Step 7 Call the resultant n chromosomes as res_pop ; Evaluate them with respect to TSCC, corresponding to every chromosome in res_pop .
- Step 8 From both par_pop and res_pop , select the best n distinct chromosomes (equal to pop_size), based on the value of TSCC, to form par_pop for the next generation.
- Step 9 Remove the best chromosome among par_pop .
- Step 10 Do the proposed local search eight times on the best chromosome, and update the best chromosome, if it improves through local search, place the same chromosome in par_pop for the next generation.
- Step 11 Increment $no_gen = no_gen + 1$;
If no_gen is ≤ 500 , then return to Step 4; else proceed to Step 12.
- Step 12 Stop. The best solution (i.e., the best chromosome) among the chromosomes in the final par_pop and its TSCC constitute the solution to the problem.

5 Performance Analysis of (s, S) Policy Under Consideration

As mentioned earlier, this paper attempts to determine best base-stocks and re-order points with (s, S) policy in a serial SC operating with deterministic demands over a finite planning horizon and make a comparative evaluation between policies, namely, periodic-review order-up-to S policy (i.e., (T, S) policy), and (s, S) policy. As for the determination of the optimal order-up-to S policy, Sethupathi and Rajendran (2010) proposed a mathematical programming model that can be executed with $T = 1200$ days, a substantive long time horizon. In view of the limitation in executing the proposed mathematical programming models for (s, S) policy over such a long time horizon, we present three GAs that are not

constrained by such limitations even though GAs cannot guarantee optimal solutions in the respective class of inventory order policies. We can therefore compare the optimal solution from the mathematical model for order-up-to S policy and heuristic solutions from GAs in order to have an idea about the relative evaluation of the respective order policies in a serial SC.

As for SC settings, we consider the same settings presented in Sect. 3.4, with T set to 1200 days. Before embarking on this relative evaluation of control policies, we first present the details of a pilot study involving parameter settings for the MGGGA in respect of (s, S) policy. We wish to state that similar observations with respect to GA parameters in HGA1 and HGA2 have been made in the case of (s, S) policy and all these details of the pilot study are not presented here for saving space.

5.1 Parameter Settings for GAs

The crossover rate CR is set to one in this study because the crossover operators are population-based crossover operators and we generate as many offspring as the number of parent chromosomes. For fixing the MR and parameter x which are used in the mutation operator, we conduct a pilot study with the consideration of the MGGGA and with lead time setting LT1. As for (MR, x) , we have experimented respectively with the values of $(0.1, 0.1)$, $(0.2, 0.1)$, $(0.1, 0.2)$, and $(0.2, 0.2)$; we have chosen these values as in most experiments involving GAs, MR is usually small and we have set x not to exceed 0.2 in order to search in a limited neighborhood. The TSCC values corresponding to these parameter settings with their relative percentage deviations from the best solution among them are evaluated in a pilot study. After evaluating the four settings, we have found that the setting $(MR = 0.2, x = 0.2)$ performs the best.

As for fixing the probability of merging P_m and merging rate R_m for the case of HGA2, we have conducted a pilot study with the values of $(0.2, 0.25)$, $(0.2, 0.3)$, $(0.25, 0.25)$, and $(0.25, 0.3)$. The reasoning of these settings are as follows: since there are four genes corresponding to the base-stocks/re-order points at four installations, the probability of altering a gene's value (obtained by gene-wise crossover operator) is not to exceed 0.25; and the relative importance given to the gene values obtained by arithmetic crossover operator is set not to exceed one-third in relation to the gene-wise crossover operator as otherwise we may not have a diversity in the generated offspring. The TSCC values corresponding to the lead time setting LT1 with their relative percentage deviations from the best solution among them are evaluated. After evaluating the four settings, we have found that the setting $(P_m = 0.25, \text{ and } R_m = 0.3)$ performs the best.

5.2 Performance of Local Search Technique in the Search Process

We have employed a local search technique to search in the neighborhood of the best solution at the end of every generation in our GAs. The introduction of local search into the GA mechanism is seen to hasten the convergence and hence the local search technique appears effective.

5.3 Run Length Concerning the Execution of GAs

Almost all the three GAs converge within 50 generations which is the common phenomenon observed over all SC settings. However, we executed the GAs over 500 generations because the CPU time is in the order of few seconds only.

5.4 Results of Execution of GAs for (s, S) Policy

The GAs for the (s, S) policy have been executed with the consideration of various SC and lead time settings with $T = 1200$ days, and with two customer demand streams, namely, first demand stream (FD) and antithetic demand stream (AD). Tables 4, 5, 6, 7, 8, and 9 show the consolidated results obtained with the respective TSCC values with best base-stocks and re-order points for all members by the three GA variants, and their performance comparison. Tables 4 and 5 show the performance of the proposed GA variants with respective TSCC values, and base-stocks and re-order points in respect of lead time setting 1 (LT1) with FD and AD, respectively. Tables 6 and 7 show the performance of the GA variants with respective TSCC values, and base-stocks and re-order points in respect of lead time setting 2 (LT2) with the FD and AD demand stream, respectively. Tables 8 and 9 show the performance of the proposed GA variants with respective TSCC values, and base-stocks and re-order points in respect of lead time setting 3 (LT3) with the FD and AD demand stream, respectively.

The relative percentage deviation for a given GA variant's solution from the best GA solution is calculated as follows:

$$\text{Relative percentage deviation} = \left(\frac{\{(\text{TSCC}_{\text{GA}} - \text{TSCC}_{\text{best}})/(\text{TSCC}_{\text{best}})\}} \right) \times 100 \quad (26)$$

where TSCC_{GA} is the TSCC value obtained by the respective GA variant and $\text{TSCC}_{\text{best}}$ is the best TSCC obtained from the three GA variants for the particular cost and lead time settings. The computational time taken by GA variants are also reported in the tables. For every SC setting and lead time setting, the number of

Table 4 Performance of the different approaches, in respect of TSSC values, base-stocks and re-order points at different members with the lead time setting I (LTI) with the first demand stream (FD) using the (s, S) policy

Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time(in seconds)	SOL En.	Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time(in seconds)	SOL En.
CS1 (LTI)	MGGA	{56, 125, 65, 84}	{53, 10, 32, 21}	1262474	2.42	14.13	10272	CS9	MGGA	{23, 102, 66, 164}	{12, 96, 5, 0}	2420490	0	2.45	1488
	HGA1	{55, 52, 62, 89}	{53, 40, 27, 15}	1232615	0	5.93	4128	(LTI)	HGA1	{23, 102, 66, 164}	{10, 81, 61, 0}	2420490	0	5.38	3648
	HGA2	{56, 125, 65, 84}	{32, 10, 14, 21}	1262474	2.42	15.00	10992		HGA2	{27, 108, 64, 179}	{14, 85, 5, 0}	2420700	0.01	1.45	768
CS2 (LTI)	MGGA	{78, 78, 79, 86}	{76, 61, 73, 23}	1449759	0	2.12	1248	CS10	MGGA	{76, 117, 82, 164}	{74, 110, 3, 0}	2918794	0.33	11.31	7968
	HGA1	{78, 78, 79, 86}	{68, 66, 41, 23}	1449759	0	2.57	1440	(LTI)	HGA1	{70, 113, 77, 159}	{36, 88, 5, 9}	2909248	0	4.56	2880
	HGA2	{78, 64, 79, 88}	{48, 1, 41, 25}	1455079	0.37	4.64	2256		HGA2	{71, 112, 78, 158}	{49, 61, 6, 8}	2909319	0	17.01	12384
CS3 (LTI)	MGGA	{92, 92, 92, 95}	{90, 85, 83, 34}	1579650	0	6.99	4896	CS11	MGGA	{155, 155, 155, 140}	{78, 147, 16, 0}	3349598	4.64	6.66	4512
	HGA1	{92, 92, 92, 95}	{43, 86, 75, 34}	1579650	0	3.96	2256	(LTI)	HGA1	{135, 135, 135, 138}	{125, 107, 44, 25}	3201355	0.01	4.41	2880
	HGA2	{92, 92, 92, 90}	{56, 51, 84, 27}	1594189	0.92	2.24	1344		HGA2	{135, 135, 135, 139}	{105, 61, 46, 26}	3201091	0	14.44	10320
CS4 (LTI)	MGGA	{100, 100, 100, 102}	{98, 79, 92, 41}	1668764	0	3.83	2496	CS12	MGGA	{145, 145, 145, 148}	{74, 144, 129, 35}	3312095	0	6.77	4608
	HGA1	{100, 100, 100, 100}	{94, 78, 88, 39}	1670508	0.10	3.48	2208	(LTI)	HGA1	{145, 145, 145, 148}	{139, 111, 48, 35}	3312095	0	5.13	3264
	HGA2	{175, 100, 100, 102}	{75, 99, 51, 41}	1698479	1.78	2.65	1392		HGA2	{270, 146, 145, 148}	{111, 88, 54, 35}	3330771	0.56	20.73	10992
CS5 (LTI)	MGGA	{29, 73, 72, 127}	{14, 67, 36, 0}	1731981	0.05	2.71	1680	CS13	MGGA	{128, 73, 56, 257}	{127, 56, 27, 0}	3384284	0	3.19	2064
	HGA1	{29, 69, 65, 121}	{16, 65, 4, 8}	1731087	0	10.47	7008	(LTI)	HGA1	{47, 314, 63, 239}	{22, 20, 28, 0}	3395575	0.33	10.26	7200
	HGA2	{31, 75, 68, 131}	{30, 48, 30, 4}	1731175	0.01	11.71	8400		HGA2	{128, 73, 56, 257}	{124, 39, 6, 0}	3384284	0	3.77	2496
CS6 (LTI)	MGGA	{84, 84, 85, 94}	{79, 65, 42, 18}	2085400	0	6.32	4368	CS14	MGGA	{60, 188, 73, 231}	{56, 147, 36, 0}	3970360	0	2.43	1440
	HGA1	{84, 84, 85, 94}	{78, 65, 37, 18}	2085400	0	5.71	3888	(LTI)	HGA1	{61, 187, 74, 231}	{43, 103, 34, 0}	3970370	0	4.38	2880
	HGA2	{84, 84, 85, 94}	{82, 46, 35, 18}	2085400	0	2.43	1488		HGA2	{61, 187, 74, 231}	{58, 100, 5, 0}	3970370	0	4.06	2688
CS7 (LTI)	MGGA	{92, 92, 92, 96}	{86, 86, 85, 34}	2250332	0.52	7.50	4368	CS15	MGGA	{198, 198, 198, 190}	{121, 121, 67, 0}	4624960	1.31	5.65	3744
	HGA1	{92, 92, 92, 96}	{85, 73, 75, 33}	2250277	0.52	3.05	1920	(LTI)	HGA1	{177, 177, 177, 181}	{112, 114, 45, 17}	4566792	0.04	3.81	2448
	HGA2	{98, 98, 98, 102}	{72, 58, 34, 28}	2238603	0	5.89	2976		HGA2	{190, 190, 190, 197}	{100, 173, 32, 18}	4565050	0	6.29	4320
CS8 (LTI)	MGGA	{102, 102, 102, 103}	{99, 79, 90, 40}	2341597	0	6.79	4656	CS16	MGGA	{411, 214, 214, 190}	{60, 41, 74, 0}	5056339	7.15	3.68	2304
	HGA1	{102, 102, 102, 103}	{100, 78, 91, 40}	2341597	0	4.74	3168	(LTI)	HGA1	{362, 362, 192, 193}	{175, 130, 62, 29}	4872679	3.26	2.83	1728
	HGA2	{103, 103, 103, 101}	{58, 90, 41, 36}	2350979	0.40	7.13	4896		HGA2	{204, 204, 205, 206}	{103, 108, 55, 27}	4719010	0	4.20	2352

[S, M, D, R] (Supplier, Manufacturer, Distributor, Retailer), MGGA Modified Gene-wise Genetic Algorithm, HGA Hybrid Genetic Algorithm-1, HGA2 Hybrid Genetic Algorithm-2, SOL En. Number of solutions enumerated to find the best solution, Rel. % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds) / CPU time required to obtain the best solution

Table 5 Performance of the different approaches, in respect of TSSC values, base-stocks and re-order points at different members with the lead time setting I (LTI) with the antithetic demand stream (AD) using the (s, S) policy

Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.	Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.
CS1 (LTI)	MGGA	[56, 47, 62, 83]	[28, 44, 29, 19]	1240754	0	5.80	3744	CS9	MGGA	[26, 105, 64, 167]	[13, 99, 32, 0]	2416464	0	2.40	1440
	HGA1	[58, 46, 66, 81]	[43, 43, 25, 14]	1242656	0.15	5.02	3408	(LTI)	HGA1	[26, 105, 64, 167]	[15, 81, 31, 0]	2416464	0	2.82	1728
CS2 (LTI)	HGA2	[57, 116, 66, 82]	[35, 13, 27, 18]	1269382	2.31	8.66	8448	CS10 (LTI)	MGGA	[26, 104, 65, 167]	[15, 62, 8, 0]	2416464	0	1.83	1392
	MGGA	[76, 76, 78, 88]	[74, 60, 69, 24]	1458483	0	3.13	1872		HGA1	[75, 113, 83, 159]	[69, 104, 5, 0]	2914694	0.30	4.93	3312
CS3 (LTI)	HGA1	[76, 76, 78, 87]	[73, 45, 22, 23]	1459043	0.04	2.95	1872	CS11 (LTI)	HGA1	[67, 119, 75, 165]	[60, 92, 37, 7]	2905845	0	3.19	2016
	HGA2	[76, 76, 78, 88]	[65, 39, 27, 24]	1458483	0	2.01	1440		HGA2	[67, 119, 75, 165]	[63, 83, 18, 7]	2905845	0	4.53	4080
CS4 (LTI)	MGGA	[86, 86, 86, 89]	[80, 67, 77, 34]	1586585	0	5.51	3696	CS12 (LTI)	MGGA	[155, 155, 154, 139]	[143, 146, 15, 0]	3341346	4.50	16.90	12048
	HGA1	[86, 86, 86, 89]	[84, 66, 75, 34]	1586585	0	3.28	2112		HGA1	[142, 142, 142, 144]	[105, 38, 75, 21]	3197606	0	2.69	1584
CS5 (LTI)	HGA2	[86, 86, 86, 88]	[45, 47, 40, 33]	1586921	0.02	4.72	2400	CS13 (LTI)	HGA2	[142, 142, 142, 140]	[82, 108, 26, 15]	3210656	0.41	7.11	6720
	MGGA	[94, 94, 94, 94]	[92, 72, 85, 39]	1671642	0	3.48	2208		MGGA	[145, 145, 145, 147]	[63, 113, 129, 33]	3321356	0	11.68	7056
CS6 (LTI)	HGA1	[94, 94, 94, 94]	[51, 74, 86, 39]	1671642	0	2.45	1488	CS14 (LTI)	HGA1	[145, 145, 145, 147]	[50, 111, 44, 33]	3321356	0	6.51	4416
	HGA2	[102, 102, 102, 103]	[67, 56, 42, 39]	1699639	1.67	1.95	1536		HGA2	[269, 145, 145, 147]	[107, 79, 58, 33]	3342001	0.62	22.66	22128
CS7 (LTI)	MGGA	[26, 183, 77, 114]	[10, 10, 37, 0]	1767358	2.19	13.25	9120	CS15 (LTI)	MGGA	[60, 80, 56, 272]	[31, 76, 27, 0]	3379794	0	2.81	1728
	HGA1	[27, 73, 69, 123]	[15, 57, 32, 2]	1729407	0	3.82	2496		HGA1	[47, 306, 62, 234]	[21, 24, 19, 0]	3393434	0.40	10.58	7200
CS8 (LTI)	HGA2	[29, 72, 70, 121]	[27, 67, 4, 0]	1731265	0.11	1.97	1440	CS16 (LTI)	HGA2	[60, 80, 56, 272]	[55, 63, 12, 0]	3379794	0	1.89	1392
	MGGA	[86, 86, 87, 98]	[81, 52, 43, 18]	2087041	0.08	8.19	4368		MGGA	[60, 190, 72, 234]	[58, 150, 36, 0]	3964442	0	3.32	2112
CS9 (LTI)	HGA1	[86, 86, 87, 98]	[82, 67, 38, 18]	2087041	0.08	3.04	1824	CS17 (LTI)	HGA1	[61, 189, 72, 234]	[47, 118, 25, 0]	3964430	0	5.18	3456
	HGA2	[91, 118, 92, 99]	[80, 8, 42, 19]	2085283	0	3.05	2688		HGA2	[61, 189, 72, 234]	[28, 114, 4, 0]	3964430	0	8.82	8448
CS10 (LTI)	MGGA	[94, 94, 94, 97]	[92, 72, 84, 30]	2252113	0.02	8.95	6288	CS18 (LTI)	MGGA	[190, 190, 190, 180]	[186, 116, 55, 0]	4620248	1.27	3.21	2016
	HGA1	[108, 108, 108, 112]	[105, 85, 59, 30]	2255693	0.18	2.69	1632		HGA1	[190, 190, 190, 195]	[134, 10, 25, 15]	4562360	0	16.38	11616
CS11 (LTI)	HGA2	[101, 101, 101, 105]	[58, 62, 56, 29]	2251563	0	4.18	2064	CS19 (LTI)	HGA2	[190, 190, 190, 195]	[114, 102, 45, 15]	4562360	0	3.28	2784
	MGGA	[103, 103, 103, 104]	[102, 81, 92, 37]	2360495	0	9.47	6576		MGGA	[219, 219, 218, 194]	[80, 218, 138, 0]	5000461	6.0	3.15	1968
CS12 (LTI)	HGA1	[183, 102, 102, 104]	[63, 55, 90, 39]	2371349	0.46	5.09	3408	CS20 (LTI)	HGA1	[362, 191, 191, 193]	[135, 189, 30, 30]	4738642	0.41	9.13	6288
	HGA2	[117, 117, 117, 114]	[80, 87, 56, 32]	2376583	0.68	5.90	5472		HGA2	[206, 206, 206, 205]	[115, 107, 68, 25]	4719524	0	7.79	7296

[S, M, D, R] (Supplier, Manufacturer, Distributor, Retailer), MGGA Modified Gene-wise Genetic Algorithm, HGA Hybrid Genetic Algorithm-1, HGA2 Hybrid Genetic Algorithm-2, SOL En. Number of solutions enumerated to find the best solution, Rel. % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds) / CPU time required to obtain the best solution

Table 6 Performance of the different approaches, in respect of TSSC values, base-stocks and re-order points at different members with the lead time setting 2 (LT2) with the first demand stream (FD) using the (s, S) policy

Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.	Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.
CS1 (LT2)	MGGA	{95, 97, 95, 121}	{93, 76, 85, 55}	1160293	0	4.25	2736	CS9	MGGA	{132, 81, 148, 186}	{121, 64, 16, 0}	2490912	2.71	13.88	8976
	HGA1	{95, 97, 95, 121}	{85, 75, 48, 55}	1160293	0	5.44	3744	(LT2)	HGA1	{116, 80, 122, 195}	{77, 65, 65, 31}	2425106	0	19.66	14208
CS2 (LT2)	HGA2	{94, 98, 96, 120}	{74, 48, 84, 52}	1161689	0.12	2.68	2208		HGA2	{115, 81, 122, 195}	{74, 42, 52, 31}	2425116	0	19.61	19584
	MGGA	{104, 107, 105, 135}	{97, 86, 93, 61}	1292146	2.25	3.45	2160	CS10	MGGA	{143, 146, 140, 174}	{139, 115, 47, 51}	2805508	0	7.05	4464
CS3 (LT2)	HGA1	{189, 105, 106, 130}	{68, 83, 86, 59}	1324509	4.81	3.84	2400	(LT2)	HGA1	{143, 146, 140, 174}	{71, 113, 38, 51}	2805508	0	5.16	3456
	HGA2	{98, 100, 99, 130}	{47, 50, 73, 69}	1263679	0	19.12	19392		HGA2	{143, 146, 140, 170}	{105, 78, 47, 47}	2809068	0.13	2.89	2400
CS4 (LT2)	MGGA	{109, 110, 109, 142}	{107, 66, 101, 71}	1392565	2.53	3.05	1632	CS11	MGGA	{213, 213, 214, 152}	{77, 119, 73, 68}	3235280	9.48	12.11	6720
	HGA1	{103, 103, 103, 141}	{62, 97, 80, 80}	1358145	0	2.66	1584	(LT2)	HGA1	{150, 151, 150, 183}	{148, 114, 74, 64}	2955200	0	4.27	2784
CS5 (LT2)	HGA2	{103, 103, 103, 141}	{54, 93, 54, 80}	1358145	0	3.55	3168		HGA2	{275, 148, 144, 181}	{111, 82, 82, 68}	2985629	1.03	3.56	2784
	MGGA	{116, 115, 116, 148}	{113, 89, 104, 74}	1505806	5.38	15.90	9216	CS12	MGGA	{351, 186, 186, 216}	{154, 183, 83, 67}	3224856	0	18.71	10080
CS6 (LT2)	HGA1	{107, 107, 107, 148}	{105, 99, 94, 87}	1428967	0	2.73	1680	(LT2)	HGA1	{252, 253, 253, 175}	{122, 169, 103, 79}	3391258	5.16	4.44	2928
	HGA2	{280, 107, 107, 148}	{73, 82, 67, 87}	1509804	5.66	9.12	4416		HGA2	{228, 228, 229, 162}	{108, 120, 75, 78}	3386104	5.00	5.92	3072
CS7 (LT2)	MGGA	{78, 124, 170, 127}	{40, 117, 17, 56}	1843996	5.58	7.25	4512	CS13	MGGA	{29, 135, 139, 257}	{16, 104, 124, 0}	3420768	0.13	2.66	1584
	HGA1	{110, 112, 115, 141}	{69, 78, 48, 44}	1748986	0.14	3.51	2256	(LT2)	HGA1	{29, 135, 139, 257}	{16, 104, 23, 0}	3420768	0.13	2.89	1632
CS8 (LT2)	HGA2	{109, 112, 112, 145}	{67, 100, 97, 49}	1746523	0	4.53	4080		HGA2	{29, 143, 130, 265}	{28, 85, 34, 81}	3416216	0	3.94	3408
	MGGA	{113, 114, 113, 144}	{111, 89, 101, 60}	1900822	0	15.57	9360	CS14	MGGA	{201, 77, 203, 223}	{55, 45, 31, 0}	4291938	3.94	8.45	5616
CS9 (LT2)	HGA1	{170, 187, 105, 135}	{116, 75, 94, 61}	1986028	4.48	6.97	4800	(LT2)	HGA1	{192, 192, 192, 224}	{108, 165, 24, 36}	4129102	0	3.16	1968
	HGA2	{113, 114, 113, 144}	{56, 88, 85, 60}	1900822	0	19.21	19104		HGA2	{192, 192, 192, 223}	{135, 112, 65, 35}	4129422	0.01	3.02	2352
CS10 (LT2)	MGGA	{121, 122, 120, 154}	{117, 95, 110, 70}	2028180	0	3.74	2016	CS15	MGGA	{192, 192, 192, 225}	{118, 34, 28, 58}	4328618	0	26.94	18816
	HGA1	{121, 122, 120, 153}	{53, 114, 100, 69}	2028228	0	2.99	1872	(LT2)	HGA1	{192, 192, 192, 225}	{85, 108, 81, 58}	4328618	0	12.40	8736
CS11 (LT2)	HGA2	{197, 111, 111, 142}	{79, 64, 43, 68}	2032496	0.21	15.19	7968		HGA2	{214, 214, 214, 236}	{140, 45, 70, 48}	4356324	0.64	4.31	1824
	MGGA	{222, 129, 126, 161}	{51, 93, 79, 78}	2175005	0	6.94	3120	CS16	MGGA	{402, 207, 213, 249}	{180, 30, 152, 72}	452478	1.61	13.18	9120
CS12 (LT2)	HGA1	{197, 197, 199, 151}	{121, 137, 88, 78}	2357094	8.37	28.62	20784	(LT2)	HGA1	{377, 198, 198, 235}	{167, 196, 84, 71}	4481540	0.66	16.31	11616
	HGA2	{181, 182, 110, 150}	{112, 90, 48, 84}	2204878	1.37	9.30	8784		HGA2	{198, 198, 198, 235}	{105, 185, 63, 71}	4452336	0	5.49	4944

{S, M, D, R} [Supplier, Manufacturer, Distributor, Retailer], % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds)/CPU time required to obtain the best solution enumerated to find the best solution, Rel. % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds)/CPU time required to obtain the best solution

Table 7 Performance of the different approaches, in respect of TSSC values, base-stocks and re-order points at different members with the lead time setting 2 (LT2) with the antithetic demand stream (AD) using the (s, S) policy

Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.	Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.
CS1 (LT2)	MGGA	[94, 98, 95, 118]	[92, 77, 87, 53]	1159704	0	3.04	1920	CS9	MGGA	[132, 80, 148, 187]	[123, 75, 15, 0]	2483804	2.51	2.40	1392
	HGA1	[94, 98, 96, 117]	[87, 76, 86, 52]	1160244	0.05	2.63	1584	(LT2)	HGA1	[111, 86, 117, 195]	[104, 65, 98, 31]	2423126	0.01	12.56	8880
CS2 (LT2)	MGGA	[94, 98, 95, 118]	[93, 91, 32, 53]	1159704	0	2.47	1920	HGA2	HGA2	[112, 85, 118, 194]	[80, 50, 45, 30]	2422974	0	12.97	12576
	HGA1	[103, 105, 103, 131]	[101, 84, 94, 59]	291230	1.56	3.23	1920	MGGA	MGGA	[179, 239, 126, 161]	[176, 65, 115, 59]	2940405	4.87	15.14	10368
CS3 (LT2)	MGGA	[100, 101, 100, 128]	[96, 80, 87, 64]	1271446	0	3.21	2016	(LT2)	HGA1	[143, 145, 141, 174]	[90, 113, 87, 49]	2803990	0	4.52	2976
	HGA1	[103, 102, 102, 125]	[53, 65, 90, 58]	1284596	1.03	4.56	4128	HGA2	HGA2	[143, 145, 141, 174]	[93, 84, 54, 49]	2803990	0	13.82	13584
CS4 (LT2)	MGGA	[106, 106, 106, 134]	[100, 70, 95, 67]	1397560	2.18	6.39	4272	CS11	MGGA	[210, 210, 215, 149]	[131, 142, 74, 68]	3206676	7.99	3.55	2064
	HGA1	[171, 179, 104, 139]	[97, 80, 81, 78]	1495958	9.38	2.63	1584	(LT2)	HGA1	[251, 139, 135, 170]	[91, 104, 122, 68]	3009665	1.36	16.74	11232
CS5 (LT2)	MGGA	[105, 105, 104, 140]	[97, 104, 64, 76]	1367716	0	2.49	2064	HGA2	HGA2	[146, 147, 144, 181]	[92, 82, 124, 67]	2969288	0	2.92	2208
	HGA1	[114, 114, 115, 143]	[111, 80, 101, 71]	1532334	5.38	3.45	2064	CS12	MGGA	[415, 224, 225, 161]	[185, 150, 92, 79]	3434724	9.84	31.25	22512
CS6 (LT2)	MGGA	[109, 108, 109, 147]	[107, 101, 96, 83]	1454086	0	11.70	8352	(LT2)	HGA1	[250, 250, 251, 173]	[183, 134, 87, 78]	3401776	8.79	3.18	2256
	HGA1	[109, 109, 109, 147]	[66, 69, 63, 83]	1454086	0	2.92	2064	HGA2	HGA2	[258, 144, 143, 179]	[115, 96, 90, 77]	3126982	0	7.13	5760
CS7 (LT2)	MGGA	[109, 113, 110, 143]	[100, 103, 102, 48]	1747675	0	20.15	14064	CS13	MGGA	[32, 137, 138, 259]	[18, 109, 126, 0]	3427836	0.15	3.20	2016
	HGA1	[109, 113, 111, 142]	[91, 85, 48, 47]	1747723	0	2.36	1392	(LT2)	HGA1	[40, 142, 125, 262]	[28, 106, 26, 12]	3422756	0	2.97	1824
CS8 (LT2)	MGGA	[108, 114, 110, 143]	[67, 93, 79, 48]	1747672	0	7.99	7728	HGA2	HGA2	[101, 138, 121, 255]	[94, 78, 59, 16]	3427866	0.15	12.31	12000
	HGA1	[113, 114, 113, 144]	[111, 90, 101, 62]	1890800	0	5.56	3696	CS14	MGGA	[76, 358, 196, 191]	[73, 119, 21, 0]	4323456	4.89	35.00	23376
CS9 (LT2)	MGGA	[118, 117, 116, 143]	[105, 104, 89, 55]	1906080	0.42	2.88	1776	(LT2)	HGA1	[192, 191, 191, 223]	[152, 150, 27, 36]	4121822	0	2.63	1584
	HGA1	[113, 114, 113, 144]	[88, 60, 76, 62]	1890800	0	2.17	1728	HGA2	HGA2	[191, 191, 191, 222]	[134, 103, 45, 35]	4122182	0.01	2.21	1728
CS10 (LT2)	MGGA	[118, 118, 118, 149]	[115, 117, 108, 68]	2017472	0	5.79	3744	CS15	MGGA	[188, 188, 188, 224]	[140, 82, 26, 60]	4336532	0.06	17.97	11280
	HGA1	[217, 120, 118, 150]	[80, 92, 50, 69]	2039543	1.09	10.05	7056	(LT2)	HGA1	[194, 194, 194, 227]	[69, 34, 87, 60]	4333738	0	7.70	5280
CS11 (LT2)	MGGA	[181, 105, 104, 139]	[78, 104, 85, 75]	2045619	1.40	13.53	13392	HGA2	HGA2	[211, 211, 211, 235]	[141, 120, 70, 48]	4338776	0.12	4.93	4512
	HGA1	[196, 196, 193, 150]	[192, 152, 79, 80]	2386342	11.11	28.80	20784	CS16	MGGA	[216, 216, 212, 252]	[134, 145, 32, 72]	4496444	0.58	21.28	13824
CS12 (LT2)	MGGA	[196, 196, 197, 150]	[117, 172, 83, 80]	2380510	10.84	7.97	7632	(LT2)	HGA1	[372, 199, 199, 234]	[167, 192, 72, 70]	4496856	0.59	9.29	6240
	HGA1	[130, 130, 130, 161]	[127, 123, 118, 73]	2147752	0	3.23	1824	HGA2	HGA2	[202, 202, 202, 238]	[93, 119, 108, 71]	4470316	0	8.00	7488

[S, M, D, R] [Supplier, Manufacturer, Distributor, Retailer], MGGA Modified Gene-wise Genetic Algorithm, HGA Hybrid Genetic Algorithm-1, HGA2 Hybrid Genetic Algorithm-2, SOL En. Number of solutions enumerated to find the best solution, Rel. % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds) CPU time required to obtain the best solution

Table 8 Performance of the different approaches, in respect of TSSC values, base-stocks and re-order points at different members with the lead time setting 3 (LT3) with the first demand stream (FD) using the (s, S) policy

Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.	Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. % dev.	CPU time (in seconds)	SOL En.
CS1 (LT3)	MGGA	{141, 143, 141, 193}	{130, 113, 122, 80}	1131416	0	13.05	9408	CS9	MGGA	{158, 158, 160, 201}	{146, 96, 101, 64}	2398797	0.30	6.82	4512
	HGA1	{141, 144, 143, 175}	{84, 110, 30, 62}	1164298	2.91	14.10	7488	(LT3)	HGA1	{157, 159, 157, 210}	{138, 134, 72, 73}	2391652	0	3.79	2016
CS2 (LT3)	MGGA	{141, 143, 141, 193}	{81, 89, 100, 80}	1131416	0	10.53	10464		HGA2	{157, 159, 157, 210}	{99, 89, 86, 73}	2391652	0	7.01	6576
	HGA1	{148, 147, 147, 207}	{138, 98, 94, 94}	1282426	0	7.94	4368	CS10	MGGA	{151, 152, 150, 210}	{140, 122, 95, 91}	2549510	0	13.33	7632
CS3 (LT3)	MGGA	{148, 147, 147, 207}	{136, 138, 55, 94}	1282426	0	4.10	2160	(LT3)	HGA1	{178, 179, 177, 226}	{70, 109, 87, 76}	2622960	2.88	7.21	4944
	HGA1	{148, 147, 147, 207}	{111, 84, 63, 94}	1282426	0	12.76	12480		HGA2	{166, 166, 164, 223}	{61, 95, 93, 88}	2578870	1.15	2.76	2304
CS4 (LT3)	MGGA	{288, 156, 155, 217}	{130, 148, 99, 101}	1511206	6.76	3.42	1920	CS11	MGGA	{156, 157, 157, 221}	{97, 124, 144, 102}	2683438	0.06	2.76	1488
	HGA1	{285, 152, 153, 211}	{119, 92, 139, 98}	1509697	6.65	12.98	9120	(LT3)	HGA1	{156, 157, 156, 222}	{90, 113, 96, 103}	2681954	0	3.76	2256
CS5 (LT3)	MGGA	{152, 152, 152, 219}	{81, 89, 80, 106}	1415521	0	2.31	1584		HGA2	{293, 158, 156, 226}	{126, 103, 95, 107}	2709617	1.03	4.44	3984
	HGA1	{297, 159, 158, 231}	{134, 151, 99, 115}	1610888	6.02	3.52	2016	CS12	MGGA	{177, 178, 177, 251}	{108, 107, 126, 116}	2847567	1.26	3.36	2112
CS6 (LT3)	MGGA	{161, 161, 160, 234}	{99, 95, 70, 115}	1545372	1.71	11.51	7968	(LT3)	HGA1	{203, 203, 202, 279}	{114, 122, 149, 115}	2935495	4.39	3.78	2448
	HGA1	{155, 155, 155, 233}	{89, 93, 101, 120}	1519424	0	12.44	12336		HGA2	{298, 161, 160, 236}	{129, 141, 76, 117}	2812082	0	3.22	2784
CS7 (LT3)	MGGA	{141, 143, 141, 193}	{138, 112, 129, 80}	1559816	0	16.89	10272	CS13	MGGA	{189, 101, 196, 285}	{112, 79, 21, 28}	3400310	0.49	2.31	1392
	HGA1	{141, 143, 142, 191}	{123, 110, 82, 78}	1560348	0.03	2.60	1584	(LT3)	HGA1	{172, 118, 178, 303}	{154, 50, 111, 46}	3383694	0	42.17	23664
CS8 (LT3)	MGGA	{141, 143, 141, 193}	{104, 71, 81, 80}	1559816	0	7.71	7488		HGA2	{169, 121, 176, 305}	{106, 81, 43, 48}	3384230	0.02	23.55	23568
	HGA1	{148, 147, 147, 207}	{136, 116, 92, 94}	1710826	0	5.09	3168	CS14	MGGA	{201, 203, 199, 261}	{124, 124, 59, 82}	3878515	0	4.78	3168
CS9 (LT3)	MGGA	{152, 154, 153, 190}	{80, 153, 32, 69}	1825025	6.68	41.50	23184	(LT3)	HGA1	{201, 203, 199, 259}	{103, 56, 64, 80}	3878835	0.01	9.72	6720
	HGA1	{148, 147, 147, 207}	{69, 83, 95, 94}	1710826	0	12.60	12480		HGA2	{201, 203, 199, 261}	{129, 104, 53, 82}	3878515	0	6.43	6000
CS10 (LT3)	MGGA	{152, 152, 152, 223}	{94, 85, 96, 110}	1841857	0	5.31	3168	CS15	MGGA	{230, 230, 229, 293}	{144, 217, 79, 94}	4095436	1.07	6.74	4608
	HGA1	{155, 155, 155, 217}	{96, 102, 105, 101}	1856621	0.80	10.56	7488	(LT3)	HGA1	{212, 212, 211, 272}	{139, 168, 52, 93}	4061339	0.23	7.84	5328
CS11 (LT3)	MGGA	{152, 152, 152, 223}	{91, 96, 93, 110}	1841857	0	8.04	7776		HGA2	{212, 212, 211, 279}	{96, 119, 106, 100}	4052147	0	4.90	4320
	HGA1	{159, 159, 158, 232}	{101, 125, 100, 116}	1953151	0.27	9.28	5520	CS16	MGGA	{255, 255, 255, 319}	{159, 153, 184, 101}	4288635	1.61	10.72	7488
CS12 (LT3)	MGGA	{161, 161, 161, 223}	{81, 96, 134, 104}	2024148	3.92	11.11	7584	(LT3)	HGA1	{418, 219, 218, 290}	{186, 148, 162, 111}	4220558	0	16.81	12000
	HGA1	{155, 155, 155, 233}	{85, 81, 97, 120}	1947824	0	6.90	6000		HGA2	{472, 245, 245, 311}	{216, 153, 86, 105}	4334670	2.70	18.14	17712

{S, M, D, R} [Supplier, Manufacturer, Distributor, Retailer], MGGA Modified Gene-wise Genetic Algorithm, HGA Hybrid Genetic Algorithm-1, HGA2 Hybrid Genetic Algorithm-2, SOL En. Number of solutions enumerated to find the best solution, Rel. % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds) required to obtain the best solution

Table 9 Performance of the different approaches, in respect of TSSC values, base-stocks and re-order points at different members with the lead time setting 3 (LT3) with the antithetic demand stream (AD) using the (s, S) policy

Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. dev.	CPU time (in seconds)	SOL En.	Supply chain setting	Solution technique	Base-stocks (S, M, D, R)	Re-order points (S, M, D, R)	TSSC	Rel. dev.	CPU time (in seconds)	SOL En.
CS1 (LT3)	MGGA	{141, 143, 143, 191}	{130, 111, 131, 79}	1156468	0.88	14.18	10272	CS9	MGGA	{156, 159, 157, 201}	{146, 96, 45, 64}	2387865	0.19	2.46	1488
	HGA1	{141, 143, 143, 191}	{127, 91, 127, 79}	1156468	0.88	6.87	4800	(LT3)	HGA1	{156, 159, 158, 199}	{104, 119, 23, 62}	2390349	0.30	6.59	4512
CS2 (LT3)	HGA2	{142, 143, 143, 181}	{119, 105, 46, 68}	1146424	0	6.99	3216		HGA2	{156, 159, 155, 209}	{91, 88, 47, 72}	2383241	0	5.57	4944
	MGGA	{147, 148, 147, 207}	{91, 82, 92, 95}	1327737	2.96	3.25	1968	CS10	MGGA	{260, 280, 150, 212}	{162, 123, 93, 95}	2727575	6.55	7.51	4992
CS3 (LT3)	HGA1	{147, 148, 147, 204}	{78, 115, 92, 92}	1329817	3.12	5.91	4032	(LT3)	HGA1	{288, 156, 152, 215}	{116, 119, 95, 93}	2593838	1.33	10.07	7056
	HGA2	{147, 147, 148, 201}	{106, 115, 82, 88}	1289576	0	3.42	3648		HGA2	{157, 157, 153, 216}	{110, 92, 56, 91}	2559789	0	8.84	7920
CS4 (LT3)	MGGA	{154, 154, 154, 224}	{141, 145, 127, 111}	1487303	3.36	3.04	1872	CS11	MGGA	{215, 216, 215, 265}	{134, 205, 179, 83}	2964752	9.71	5.72	3840
	HGA1	{161, 161, 161, 227}	{85, 151, 80, 106}	1501187	4.32	2.49	1488	(LT3)	HGA1	{163, 163, 162, 228}	{118, 127, 102, 103}	2702244	0	4.11	2688
CS5 (LT3)	HGA2	{155, 155, 155, 217}	{76, 91, 125, 101}	1439021	0	3.94	3312		HGA2	{174, 174, 172, 228}	{103, 119, 86, 91}	2744701	1.57	12.50	6144
	MGGA	{166, 166, 165, 240}	{102, 157, 104, 119}	1618965	5.11	3.21	2016	CS12	MGGA	{329, 177, 176, 249}	{141, 133, 111, 115}	2881811	1.64	29.12	20976
CS6 (LT3)	HGA1	{166, 166, 166, 237}	{87, 158, 99, 116}	1621473	5.27	10.76	6960	(LT3)	HGA1	{177, 177, 177, 249}	{79, 101, 147, 115}	2835294	0	3.21	2016
	HGA2	{159, 159, 158, 230}	{94, 80, 109, 114}	1540287	0	5.04	13536		HGA2	{303, 167, 165, 240}	{135, 97, 105, 119}	2865521	1.07	3.31	2832
CS7 (LT3)	MGGA	{141, 143, 143, 191}	{139, 113, 39, 79}	1588468	0	12.53	8880	CS13	MGGA	{147, 131, 151, 314}	{35, 76, 36, 84}	3480987	2.35	33.08	23856
	HGA1	{145, 148, 146, 197}	{127, 116, 96, 76}	1595345	0.43	3.54	2160	(LT3)	HGA1	{165, 125, 172, 308}	{103, 118, 48, 49}	3401291	0.01	31.81	22944
CS8 (LT3)	HGA2	{141, 143, 143, 191}	{83, 77, 94, 79}	1588468	0	5.05	4656		HGA2	{167, 123, 173, 307}	{103, 57, 38, 48}	3401117	0	5.30	4800
	MGGA	{151, 152, 149, 213}	{149, 102, 50, 96}	1759692	0.08	6.08	4176	CS14	MGGA	{208, 211, 205, 262}	{76, 200, 71, 75}	3867092	0	11.32	7728
CS9 (LT3)	HGA1	{157, 157, 155, 207}	{96, 105, 98, 82}	1777353	1.08	2.68	1680	(LT3)	HGA1	{208, 211, 205, 262}	{120, 47, 60, 75}	3867092	0	6.43	4368
	HGA2	{147, 148, 147, 209}	{116, 95, 87, 96}	1758359	0	3.13	2736		HGA2	{208, 211, 205, 262}	{119, 105, 71, 75}	3867092	0	3.82	3360
CS10 (LT3)	MGGA	{161, 161, 161, 226}	{100, 90, 103, 105}	1906283	0	3.77	1920	CS15	MGGA	{219, 219, 217, 275}	{80, 124, 73, 88}	4064763	0	6.40	4272
	HGA1	{293, 162, 161, 226}	{120, 105, 132, 105}	1964041	3.03	4.37	2784	(LT3)	HGA1	{421, 221, 217, 279}	{184, 39, 142, 92}	4106849	1.04	23.23	12624
CS11 (LT3)	HGA2	{294, 159, 157, 225}	{116, 86, 79, 108}	1964332	0.05	3.96	1632		HGA2	{219, 219, 217, 275}	{92, 83, 72, 88}	4064763	0	3.96	3504
	MGGA	{232, 169, 169, 168}	{105, 159, 76, 107}	2055043	3.26	3.33	2112	CS16	MGGA	{249, 249, 249, 307}	{154, 167, 209, 97}	4305884	1.32	6.42	4320
CS12 (LT3)	HGA1	{169, 169, 168, 233}	{163, 102, 104, 108}	2049683	0	4.17	2736	(LT3)	HGA1	{244, 244, 244, 311}	{142, 162, 174, 106}	4249953	0	8.73	5904
	HGA2	{303, 167, 166, 238}	{135, 100, 106, 117}	2084053	1.68	9.09	8784		HGA2	{468, 244, 244, 303}	{217, 168, 76, 98}	4347356	2.29	22.11	21696

{S, M, D, R} [Supplier, Manufacturer, Distributor, Retailer], % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds)/CPU time required to obtain the best solution enumerated to find the best solution, Rel. % Dev. Relative percentage deviation within the different approaches, CPU time (in seconds)/CPU time required to obtain the best solution

solutions enumerated by the respective GA variant to obtain the best solution is also given in the tables. The average relative percentage deviation with respect to every GA variant across all the SC setting and lead time setting are found, and reported in Table 10. From the results, we find that HGA2 performs very well with an average relative percentage of only 0.455 %, when compared with HGA1 with an average relative percentage of 1.18 %, and MGGA with an average relative percentage of 1.747 %.

We find that as the ratio of (b_j/h_j) increases for the given order costs at installations (e.g., CS1–CS4; CS5–CS8; CS9–CS12; CS13–CS16), re-order points mostly increase so as to increase the order frequency in order to replenish faster and hence reduce shortage costs across members in the SC. We also find that as the order costs increase across the SC for the given ratio of (b_j/h_j) (e.g., see CS1, CS5, CS9, and CS13; CS2, CS6, CS10, and CS14; and so on), base stock also increases at every installation to bring down the costs of orders at installations in the SC over the finite-time horizon.

5.5 Relative Performance Evaluation of (s, S) Policy with the Order-Up-To S Policy with Periodic Review

As for the determination of the optimal base-stock policy, Sethupathi and Rajendran (2010) proposed a mathematical programming model that can be executed with time horizon $T = 1200$ days. We compare these optimal TSCC values in the class of order-up-to S policy (i.e., (T, S) policy) with periodic review obtained through the execution of the above model in CPLEX Solver with the best TSCC values obtained through the execution of GAs for the (s, S) policy. Tables 11, 12, 13, 14, 15, and 16 show the comparison of optimal TSCC values obtained for order-up-to S policy with periodic review (Policy I) with best TSCC values obtained by GAs for (s, S) policy (Policy II) for all SC settings and their relative percentage deviations.

Table 10 Performance of the different GA approaches in terms of their average relative percentage deviations

Supply chain setting	Average relative percentage deviation		
	MGGA	HGA1	HGA2
CS1 to C16 (LT1-FD)	1.03	0.27	0.40
CS1 to C16 (LT1-AD)	0.90	0.11	0.36
CS1 to C16 (LT2-FD)	2.10	1.49	0.89
CS1 to C16 (LT2-AD)	3.20	2.03	0.17
CS1 to C16 (LT3-FD)	1.12	1.89	0.31
CS1 to C16 (LT3-AD)	2.15	1.30	0.60
Overall average	1.747	1.180	0.455

Table 11 Performance of inventory order policies, in respect of TSCC values, with lead time setting 1 (LT1) and with the first demand stream (FD)

Supply chain setting	TSCC			Relative percentage deviation	
	Policy-1	Policy-2	Best TSCC	Policy-1	Policy-2
CS1	1269796	1232615	1232615	3.02	0.00
CS2	1477956	1449759	1449759	1.94	0.00
CS3	1626552	1579650	1579650	2.97	0.00
CS4	1728088	1668764	1668764	3.55	0.00
CS5	1785700	1731087	1731087	3.15	0.00
CS6	2128808	2085400	2085400	2.08	0.00
CS7	2340970	2238603	2238603	4.57	0.00
CS8	2448088	2341597	2341597	4.55	0.00
CS9	2473186	2420490	2420490	2.18	0.00
CS10	2943948	2909248	2909248	1.19	0.00
CS11	3285595	3201091	3201091	2.64	0.00
CS12	3464999	3312095	3312095	4.62	0.00
CS13	3375814	3384284	3375814	0.00	0.25
CS14	3986046	3970360	3970360	0.40	0.00
CS15	4475403	4565050	4475403	0.00	2.00
CS16	4886498	4719010	4719010	3.55	0.00
Average relative percentage deviation				2.526	0.141

Policy-1 Order-up-to S policy with periodic review; the results from the solver for the mathematical model are reported here, *Policy-2* (s, S) policy; the best results from the three GAs are reported here

TSCC values obtained for Policy-1 are optimal in the class of order-up-to S policy with periodic review and TSCC values obtained for Policy-2 are the best heuristic values obtained across three genetic algorithms. The relative percentage deviation for a given policy is computed with respect to the best solution obtained across two policies

The relative percentage deviation of a given TSCC value for an order policy is computed with respect to the best TSCC value among two order policies for given SC and LT settings.

$$\begin{aligned}
 &\text{Relative percentage deviation} \\
 &= \left(\left\{ \left(\frac{\text{TSCC}_{\text{given policy}} - \text{TSCC}_{\text{best across two policies}}}{\text{TSCC}_{\text{best across two policies}}} \right) \right\} \times 100 \right) \tag{27}
 \end{aligned}$$

where $\text{TSCC}_{\text{given policy}}$ is the TSCC value obtained by the respective model, and $\text{TSCC}_{\text{best across two policies}}$ is the best TSCC value obtained among the three models for the given cost and lead time settings. The average relative percentage deviation with respect to two policies across all SC settings and lead time settings are found, and reported in Table 17.

Table 12 Performance of inventory order policies, in respect of TSCC values, with lead time setting 1 (LT1) and with the antithetic demand stream (AD)

Supply chain setting	TSCC			Relative percentage deviation	
	Policy-1	Policy-2	Best TSCC	Policy-1	Policy-2
CS1	1276128	1240754	1240754	2.85	0.00
CS2	1482414	1458483	1458483	1.64	0.00
CS3	1639252	1586585	1586585	3.32	0.00
CS4	1746782	1671642	1671642	4.49	0.00
CS5	1786562	1729407	1729407	3.30	0.00
CS6	2121121	2085283	2085283	1.72	0.00
CS7	2353012	2251563	2251563	4.51	0.00
CS8	2465827	2360495	2360495	4.46	0.00
CS9	2470582	2416464	2416464	2.24	0.00
CS10	2950900	2905845	2905845	1.55	0.00
CS11	3288207	3197606	3197606	2.83	0.00
CS12	3477966	3321356	3321356	4.72	0.00
CS13	3364038	3379794	3364038	0.00	0.47
CS14	3991674	3964430	3964430	0.69	0.00
CS15	4509889	4562360	4509889	0.00	1.16
CS16	4881722	4719524	4719524	3.44	0.00
Average relative percentage deviation				2.61	0.102

Table 13 Performance of inventory order policies, in respect of TSCC values, with lead time setting 2 (LT2) and with the first demand stream (FD)

Supply chain setting	TSCC			Relative percentage deviation	
	Policy-1	Policy-2	Best TSCC	Policy-1	Policy-2
CS1	1189399	1160293	1160293	2.51	0.00
CS2	1296907	1263679	1263679	2.63	0.00
CS3	1390537	1358145	1358145	2.39	0.00
CS4	1468331	1428967	1428967	2.75	0.00
CS5	1770090	1746523	1746523	1.35	0.00
CS6	1994638	1900822	1900822	4.94	0.00
CS7	2110537	2028180	2028180	4.06	0.00
CS8	2188331	2175005	2175005	0.61	0.00
CS9	2490090	2425106	2425106	2.68	0.00
CS10	2878023	2805508	2805508	2.58	0.00
CS11	3114440	2955200	2955200	5.39	0.00
CS12	3235882	3224856	3224856	0.34	0.00
CS13	3503646	3416216	3416216	2.56	0.00
CS14	4104041	4129102	4104041	0.00	0.61
CS15	4423184	4328618	4328618	2.18	0.00
CS16	4671965	4452336	4452336	4.93	0.00
Average relative percentage deviation				2.619	0.038

Table 14 Performance of inventory order policies, in respect of TSCC values, with lead time setting 2 (LT2) and with the antithetic demand stream (AD)

Supply chain setting	TSCC			Relative percentage deviation	
	Policy-1	Policy-2	Best TSCC	Policy-1	Policy-2
CS1	1185175	1159704	1159704	2.20	0.00
CS2	1300597	1271446	1271446	2.29	0.00
CS3	1407285	1367716	1367716	2.89	0.00
CS4	1497425	1454086	1454086	2.98	0.00
CS5	1770945	1747672	1747672	1.33	0.00
CS6	1996948	1898080	1898080	5.21	0.00
CS7	2127285	2017472	2017472	5.44	0.00
CS8	2217425	2147752	2147752	3.24	0.00
CS9	2490945	2422974	2422974	2.81	0.00
CS10	2875325	2803990	2803990	2.54	0.00
CS11	3113750	2969288	2969288	4.87	0.00
CS12	3255496	3126982	3126982	4.11	0.00
CS13	3504536	3422756	3422756	2.39	0.00
CS14	4106109	4121822	4106109	0.00	0.38
CS15	4414134	4333738	4333738	1.86	0.00
CS16	4678101	4470316	4470316	4.65	0.00
Average relative percentage deviation				3.050	0.024

Table 15 Performance of inventory order policies, in respect of TSCC values, with lead time setting 3 (LT3) and with the first demand stream (FD)

Supply chain setting	TSCC			Relative percentage deviation	
	Policy-1	Policy-2	Best TSCC	Policy-1	Policy-2
CS1	1118691	1131416	1118691	0.00	1.14
CS2	1272799	1282426	1272799	0.00	0.76
CS3	1411866	1415521	1411866	0.00	0.26
CS4	1529193	1519424	1519424	0.64	0.00
CS5	1598691	1559816	1559816	2.49	0.00
CS6	1752799	1710826	1710826	2.45	0.00
CS7	1891866	1841857	1841857	2.72	0.00
CS8	2009193	1947824	1947824	3.15	0.00
CS9	2476469	2391652	2391652	3.55	0.00
CS10	2687585	2549510	2549510	5.42	0.00
CS11	2851866	2681954	2681954	6.34	0.00
CS12	2969193	2812082	2812082	5.59	0.00
CS13	3470301	3383694	3383694	2.56	0.00
CS14	4007610	3878515	3878515	3.33	0.00
CS15	4254890	4052147	4052147	5.00	0.00
CS16	4447580	4220558	4220558	5.38	0.00
Average relative percentage deviation				3.038	0.135

Table 16 Performance of inventory order policies, in respect of TSCC values, with lead time setting 3 (LT3) and with the antithetic demand stream (AD)

Supply chain setting	TSCC			Relative percentage deviation	
	Policy-1	Policy-2	Best TSCC	Policy-1	Policy-2
CS1	1121140	1146424	1121140	0.00	2.26
CS2	1274896	1289576	1274896	0.00	1.15
CS3	1416888	1439021	1416888	0.00	1.56
CS4	1537614	1540287	1537614	0.00	0.17
CS5	1601140	1588468	1588468	0.80	0.00
CS6	1754896	1758359	1754896	0.00	0.20
CS7	1896888	1906283	1896888	0.00	0.50
CS8	2017614	2049683	2017614	0.00	1.59
CS9	2472191	2383241	2383241	3.73	0.00
CS10	2686637	2559789	2559789	4.96	0.00
CS11	2856888	2702244	2702244	5.72	0.00
CS12	2977614	2835294	2835294	5.02	0.00
CS13	3467880	3401117	3401117	1.96	0.00
CS14	4008291	3867092	3867092	3.65	0.00
CS15	4259830	4064763	4064763	4.80	0.00
CS16	4462191	4249953	4249953	4.99	0.00
Average relative percentage deviation				2.227	0.464

Table 17 Overall performance comparison of inventory order policies in respect of average relative percentage deviations of their TSCC from the best TSCC

Supply chain setting	Average relative percentage deviation	
	Policy-1	Policy-2
CS1 to C16 (LT1-FD)	2.526	0.141
CS1 to C16 (LT1-AD)	2.610	0.102
CS1 to C16 (LT2-FD)	2.619	0.038
CS1 to C16 (LT2-AD)	3.050	0.024
CS1 to C16 (LT3-FD)	3.038	0.135
CS1 to C16 (LT3-AD)	2.227	0.464
Overall average	2.678	0.151

6 Managerial Implications and Further Work of this Study

It is evident that on the whole, Policy-II, namely, the (s, S) policy emerges to be better with an average relative percentage deviation of 0.151 %, when compared with Policy-I namely, order-up-to S policy with periodic review with an average relative percentage deviation of 2.678 % over the SC settings. This finding is interesting in the sense that the (s, S) policy that capitalizes on the tight control by using the re-order point at every installation yields the least TSCC in most cases, as opposed to the periodic review system operating with order-up-to S policy. We observe that the (s, S) policy performs better than the (T, S) policy on the whole

over a number of SC settings. However, at relatively larger lead times with low cost settings, we find that the (T, S) policy becomes competitive. This is because of demand aggregation over large lead times coupled with low costs lead to less sensitivity with order policy. Hence, we can resort to the (T, S) policy under these circumstances. Further work can look at comparative study involving more inventory order policies such as (R, Q) policy, which the authors are investigating using approaches and algorithms similar to the ones presented here.

7 Summary

In this paper, we have considered the (s, S) policy and developed a mathematical model to determine the optimal base-stocks and re-order points in the class of (s, S) policy in a SC operating with deterministic demands over a finite time horizon in order to minimize the sum of holding costs, shortage costs, and order costs. In view of the computational complexity associated with this policy, we have presented three GAs to obtain the best heuristic base-stocks and re-order points across members in a SC. We have carried out a relative evaluation of the periodic review order-up-to S policy ((T, S) policy) and (s, S) policy by considering many SC costs and lead time settings. We have found that the (s, S) policy performs mostly better, in comparison to the periodic review order-up-to S policy.

Acknowledgments The authors thank the two reviewers for their suggestions to improve the paper. The second author gratefully acknowledges the Alexander von Humboldt Foundation for supporting him to carry out a part of this work at the University of Passau.

References

- Axsäter, S. (2006). *Inventory control* (2nd ed.). Springer: New York.
- Axsäter, S., & Rosling, K. (1993). Notes: Installation vs. echelon installation stocking policies for multi-level inventory control. *Management Science*, *39*, 1274–1280.
- Berry, L. M., Murtagh, B. A., McMahon, G. B., Sugden, S. J., & Welling, L. D. (1998). Genetic algorithms in the design of complex distribution networks. *International Journal of Physical Distribution and Logistics Management*, *28*, 377–381.
- Cheng, C. B., Liu, Z. Y., Chiu, S. W., & Cheng, C. J. (2006). Applying fuzzy logic control to inventory decision in a supply chain. *WSEAS Transactions on Systems*, *5*, 1822–1829.
- Cheung, K. L., & Zhang, S. H. (2008). Balanced and synchronized ordering in supply chains. *IIE Transactions*, *40*, 1–11.
- Clark, A. J., & Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. *Management Science*, *6*, 474–490.
- Daniel, J. S. R., & Rajendran, C. (2005a). A simulation-based genetic algorithmic approach to inventory optimization in a serial supply chain. *International Transactions in Operational Research*, *12*, 101–127.

- Daniel, J. S. R., & Rajendran, C. (2005b). Determination of base-stock levels in a serial supply chain: A simulation-based simulated annealing heuristic. *International Journal of Logistics Systems and Management*, 1, 149–186.
- Daniel, J. S. R., & Rajendran, C. (2006). Heuristic approaches to determine base-stock levels in a serial supply chain with a single objective and with multiple objectives. *European Journal of Operational Research*, 175, 566–592.
- Deo, N. (1999). *System simulation with digital computer*. New Delhi: Prentice-Hall of India Private Limited.
- Federguen, A., & Zipkin, P. (1984). Computational issues in an infinite horizon multi-echelon inventory problem with stochastic demand. *Operations Research*, 32, 818–836.
- Gallego, G., & Zipkin, P. (1999). Stock positioning and performance estimation in serial production-transportation systems. *Manufacturing and Services Operations Management*, 1, 77–88.
- Ganeshan, R., Jack, E., Magazine, M. J., & Stephens, P. (1999). A taxonomic review of supply chain management research. In S. Tayur, et al. (Eds.), *Quantitative models for supply chain management* (pp. 839–879). Boston, MA: Kluwer Academic Publishers.
- Glasserman, P., & Tayur, S. (1995). A sensitivity analysis for base-stock levels in a multi-echelon production-inventory systems. *Management Science*, 41, 263–281.
- Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*. Boston, MA: Addison-Wesley Publishing.
- Haq, A. N., & Kannan, G. (2006). Two-echelon distribution-inventory supply chain model for the bread industry using genetic algorithms. *International Journal of Logistics Systems and Management*, 2, 177–193.
- Hong, S. P., & Kim, Y. H. (2009). A genetic algorithm for joint replenishment based on the exact inventory cost. *Computers & Operations Research*, 36, 167–175.
- Johansen, S. G., & Thorstenson, A. (2008). Optimal base-stock policy for the inventory system with periodic review, backorders and sequential lead times. *International Journal of Inventory Research*, 1, 44–52.
- Kumanan, S., Venkatesan, S. P., & Kumar, J. P. (2007). Optimisation of supply chain logistics network using random search techniques. *International Journal of Logistics Systems and Management*, 3, 252–266.
- Lee, H. L., & Billington, C. (1992). Managing supply chain inventory: pitfalls and opportunities. *Sloan Management Review*, 33, 65–72.
- Lee, H. L., & Billington, C. (1993). Material management in decentralized supply chains. *Operations Research*, 41, 835–847.
- Lee, Y. H., Jeong, C. S., & Moon, C. (2002). Advanced planning and scheduling with outsourcing in manufacturing supply chain. *Computers & Industrial Engineering*, 43, 351–374.
- Love, S. F. (1979). *Inventory control*. New York: McGraw-Hill Book Company.
- Min, H., & Zhou, G. (2002). Supply chain modeling: past, present and future. *Computers & Industrial Engineering*, 43, 231–249.
- Nahmias, S. (2008). *Production and operations analysis* (6th ed.). New York: McGraw-Hill.
- Petrovic, D., Roy, R., & Petrovic, R. (1998). Modeling and simulation of a supply chain in an uncertain environment. *European Journal of Operational Research*, 109, 299–309.
- Rom, W. O., & Slotnick, S. A. (2009). Order acceptance using genetic algorithms. *Computers & Operations Research*, 36, 1758–1767.
- Sethupathi, P. V. R., & Rajendran, C. (2010). Optimal and heuristic base-stock levels and review periods in a serial supply chain. *International Journal of Logistics Systems and Management*, 7, 133–164.
- Shang, K. H., & Song, J. S. (2003). Newsvendor bounds and heuristic for optimal inventory policies in serial supply chains. *Management Science*, 49, 618–638.
- Shang, K. H., & Song, J. S. (2007). Serial supply chains with economies of scale: bounds and approximations. *Operations Research*, 55, 843–853.
- Silver, E., Pyke, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling* (3rd ed.). New York: John Wiley and Sons.

- van Houtum, G. J., Scheller-Wolf, A., & Yi, J. (2007). Optimal control of serial inventory systems with fixed replenishment intervals. *Operations Research*, *55*, 674–687.
- Wang, Q. (2011). Control policies for multi-echelon inventory systems with stochastic demand. In T.-M. Choi & T. C. E. Cheng (Eds.), *Supply chain coordination under uncertainty* (pp. 83–108). Berlin: Springer.
- Zhou, G., Min, H., & Gen, M. (2002). The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach. *Computers & Industrial Engineering*, *43*, 251–261.
- Zipkin, P. (2000). *Foundations of inventory management*. New York: McGraw-Hill.