



HAL
open science

Optimising integrated inventory policy for perishable items in a multi-stage supply chain

Alexandre Dolgui, Manoj Kumar Tiwari, Yerasani Sinjana, Sri Krishna Kumar, Young-Jun Son

► To cite this version:

Alexandre Dolgui, Manoj Kumar Tiwari, Yerasani Sinjana, Sri Krishna Kumar, Young-Jun Son. Optimising integrated inventory policy for perishable items in a multi-stage supply chain. *International Journal of Production Research*, 2018, 56 (1-2), pp.902 - 925. 10.1080/00207543.2017.1407500 . hal-01741662

HAL Id: hal-01741662

<https://hal.science/hal-01741662>

Submitted on 23 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimizing Integrated Inventory Policy for Perishable Items in a Multi-stage Supply Chain

Alexandre Dolgui, Manoj Kumar Tiwari, Yerasani Sinjana, Sri Krishna Kumar, Young Jun Son

International Journal of Production Research, 2018

Abstract

The value of perishable products is most affected by the time delays in a supply chain. A major issue is how to integrate the existing practices in production, inventory holding and distribution, besides considering the perishable nature of the products, so as to deliver an optimized policy for the perishable commodities. Standard inventory control models are often not adequate for perishable products and there is a need for a new integrated model to focus on consolidation of production, inventory and distribution processes. We develop such a mathematical model to search for an optimal integrated inventory policy for perishable items in a multi-stage supply chain. We specifically assume the exponential deterioration rate so as to be consistent with the growth rate of the micro-organisms responsible for deterioration. We propose and analyze some general properties of the model and apply it to a three-stage supply chain. We show that this integrated model which includes inventory control and fleet selection can be optimized with an evolutionary technique like genetic algorithm. A novel genetic algorithm that avoids revisits and employs a parameter-less self-adaptive mutation operator is developed. The results are compared with those obtained with CPLEX for small sized problems. We show that our model and optimization approach gives near optimal results for varied demand scenarios.

Keywords: Supply chain, perishable products, production-inventory-distribution, fleet selection, non-revisiting genetic algorithm.

1. Introduction

On the occasion of 55th volume anniversary of International Journal of Production Research, one of the very important and most cited journals in production research area, publishers taken an initiative to bring out some of the very good articles in cutting edge areas related to production management, manufacturing and logistics to resolve some of the challenging problems encountered by the industries. In this paper, we present a novel integrated inventory policy for perishable items in a multi-stage supply chain and algorithms for its optimization.

Since the setting up of the freight transportation industry, it has been facing the challenge of time-sensitive industrial and commercial practices. With the rapidly changing environment, more and more products are reflecting the characteristics of the perishability. Owing to the products with ever reducing shelf-life and the distribution network spanning the entire globe like never before, it is tempted to integrate the product lifetime with the crucial supply chain issues like inventory policy and fleet selection. If products that are held in stock deteriorate or can be ruined or destroyed, the product is said to display the characteristic of perishability. Nahmias has given excellent overviews of perishability as proposed by Brant (1975). The perishability of product is characterized with respect to the concept of product age. This was introduced by Nahmias (1982) firstly and then Jiang and Yang (2014) have suggested classification method on perishable goods.

Among the industries dealing with the perishable products and most affected by the seasonal variation of the product lifetime are the Fast Moving Consumer Goods (FMCG) and pharmaceutical industries. In the pharmaceutical industries, they would be a shipment of cartons of different tablet boxes, bottled medicines and solutions, radioactive dyes and catalysts, and other perishable laboratory supplies. Further, pharmaceutical companies face the problem of variable product decay rate owing to frequent changes in the chemical composition of the products. Inventories undergo change in storage in time they may become partially or entirely unfit for consumption. A similar trade-off between the time and the resources assumes significant importance during the military relief operations in times of natural calamities like flood,

cyclone, hurricane, etc (Wang et al, 2016). Another important task for any organization that moves cargo and people is to decide how many vehicles or platforms it needs. This is essentially a fleet mix (Langevin and Riopel, 2005) or capacity planning problem in the presence of available budget, time constraint, politics, etc.

While carrying out related research we have come across some very good related articles published in IJPR and similar other journals. Notable among the contributors are Lin and Chen (2003) where they studied dynamic allocation problem with uncertain supply. The objective was to maximize the total net profit of the strategic alliance of the perishable commodity supply chain and to determine the optimal orders for suppliers and resultant amount of perishable commodities allocated to retailers. The produced items and the production equipment deteriorations are considered by Alfares, Khursheed and Noman (2005). They developed a production-inventory model for deteriorating items and a heuristic solution algorithm to determine the production and inspection schedules. Shelf-life issues in production planning and scheduling of perishable products are discussed by Lütke Entrup et al. (2005). An optimization model integrating traceability initiatives with operation factors to achieve desired product quality and minimum impact of product recall is developed by Li and O'Brien (2009). Fleet selection algorithm was first proposed by Hauer (1971) for a single route and stationary, inelastic travel demand. In recent years, it has been combined with Vehicle Routing Problem (VRP) and solved together using a metaheuristic algorithm by Yousefikhoshbakht, Didehvar and Rahmati (2013). Boudia and Prins (2009) have considered an integrated production transportation problem. Integrating VRP with production scheduling in order to reach customers' heightened expectations was recently investigated by Fu, Aloulou and Triki (2017). They realized good performance and significant benefits of coordination. Nakandala, Lau and Shum (2017) faced a problem of making cost effective lateral transshipment decisions in perishable inventory management.

Most inventory models assume that stock items can be stored indefinitely to meet future demand. The purpose of this article is to consider a perishable product inventory model integrated with an optimal fleet

selection policy. Based on the idea of product age, using an adaptive evolutionary technique, the model about the perishable product inventory system with optimal fleet selection, under the assumptions of no backordering and no deterioration during the transportation, is presented for periodic review model under the control of the order-up to inventory strategy. But, the model is solved by using an innovative genetic algorithm (GA) that avoids revisits and employs a self-adaptive mutation operator.

In this paper, we propose an original approach where we include many characteristics of the perishable products that are applicable to FMCG, cosmetics, radioactive elements and pharmaceutical industries but not considered by other researchers. For example, our model assumes the exponential deterioration rate for the products which is consistent with the fact that in most of the cases the deterioration of the products is caused by the growth of micro-organisms which itself follows the exponential growth rate (Juneja and Marks, 2006; Mochizuki and Hattori, 1987; Martin and Felsenfeld, 1963). We incorporate in our model and objective function that covers the fleet size and mix vehicle selection policy and use it as part of the decision policy of the firm. Finally, we present a solution for a specific case of our model by using synthetic data and provide insightful results that can be used by any company under similar scenarios.

The rest of the paper is organized as follows. In Section 2, we present a comprehensive review of related literature and we argue how intermediate inventory holding stages affect the entire supply chain performance. In Section 3, we provide a conceptual framework to model the considered decision-making problem. In Section 4, we explain the proposed adaptive non-revisiting genetic algorithm. In Section 5, we give an extensive analysis and comparison of the computational results. In Section 6, we conclude the paper with an overview of our work.

2. LITERATURE SURVEY

The multi-period economic lot sizing (ELS) problem has been extensively studied by scholars (Dolgui and Proth, 2010). Major emphasis has been placed to this problem due to its applicability in many important practical scenarios. However, the researchers have been largely focused on production and inventory planning problems (Erenguc et al 1999). Scanning of numerous papers reveal that integrated

production, distribution and inventory (PID) models have not been investigated thoroughly. Pioneering work by Chandra and Fisher (1994) has shown that integration of these decisions can substantially increase the efficiency of the supply chain.

The Classical ELS model discussed by Wagner and Whitin (1958) minimizes the total production and inventory costs over a finite planning horizon with deterministic demand and incapacitated production. A formal description of its variants is given by Wolsey (1995). Recently, Kaminsky and Simchi-Levi (2003) showed that by considering special cost structures, a three-stage ELS model can be transformed into a two stage model. A few extensions of the ELS problem related to this work are given by Li et al. (2004), who studied the ELS problem with backordering and truckload discounts; and Hoesel et al. (2005), who studied serial supply chains in which PID decisions are integrated in the presence of production capacities. Manufacturers can avail various discount schemes as discussed by Benton and Park (1996). An extensive bibliography of non-deterministic lot-sizing models is presented in Aloulou et al. (2014).

Research on perishable products goes back to Chare and Schrader (1963) who proposed a deteriorating inventory model with constant rate of decay. Nahmias (1982) differentiated between the fixed and variable lifetime of inventory. The former implies usability of the perishable unit up to a given period n after which it expires. While, the latter has inventory deteriorating at a fixed or variable rate over time rendering some of the units unusable as time progresses. Hsu (2000) used deterioration rates that depend on both the stocks' ages and production period. Multi item joint replenishment model for non-instantaneous deteriorating items was recently considered by Ai (2017).

Multi-stage supply chains usually consist of distribution centers or warehouses as intermediate stages. We consider the case in which crossdocks are used in the supply chain instead of warehouses. This serves two purposes, one is that inventory costs are eliminated and the other is that the time the units take to reach the retailer reduces. A large share of Wal-Mart's success has been attributed to cross-docking. Some notable research on crossdocks includes Shaffer (2000), who analyses the implementation of crossdocking operations and Napolitano (2002), who investigates the use of crossdocking. A state of the art presentation

of synchronization problems in crossdock network management is presented in Buijs et al. (2014). The transportation problem of cross-docking network design integrated with truck-door assignments to minimize total transportation costs from suppliers to customers is considered in Küçüköğlü & Öztürk (2017).

Heuristics have been designed to tackle the problems similar to the one addressed in few papers (Bail et al, 2008; Tarantilis and Kiranoudis, 2001; Thomas and Griffin, 1996; Aggarwal and Park, 1993). Golden et al. (1984) developed several heuristics by revising the saving algorithm of Clarke and Wright (1964), the sweep algorithm of Gillett and Miller (1974) and the generalized assignment of Fisher and Jaikumar (1981). These traditional heuristics are outperformed by a new generation of heuristics, including mathematical programming based heuristics and meta-heuristics, particularly Tabu Search. Choi and Tcha (2007) developed a set covering formulation and solved its linear relaxation by column generation to obtain the bounds. Comparing these methods on the Golden et al. (1984) benchmark instances, the best solutions have been obtained by Choi and Tcha (2007). Evolutionary algorithms have also been attempted by Ochi et al. (1998) and Lima et al. (2004) with fixed costs, but with little success. Among these meta-heuristics designed for solving Fleet Size and Mix Vehicle Routing Problem (FSMVRP), better results are obtained by tabu search based algorithms, particularly those developed by Brandao (2009) and Wassan and Osman (2002). To further enhance the efficiency of GA, Ronald (1998) reports the use of Hash table to reduce the number of comparisons. However, these efforts only compare a child with the current population. It does not guarantee non-revisits. Povinelli and Feng (1999) use a small hash table to store all visited individuals. Kratica (1999) uses a small fixed size cache to store all visited individuals.

What distinguishes our work is that we focus on improving the overall cost of supply chain for perishable products, modeling the integration of supply chain functions viz. production, inventory and distribution, and incorporating specifically the exponential nature of product deterioration rate considering the cross-dock concept and two types of transportation – Half truck load (HTL) and full truck load (FTL). The references discussed earlier touch on different aspects of our study, but to the best of our knowledge, none

of those papers cover all of the topics and issues considered in our research.

Moreover, we use a novel modification of genetic algorithms for optimization. Indeed, for an NP-hard problem with a large number of discrete variables like the one addressed in this article, the evolutionary algorithms appear as the primary choice for solving it. This is attributed to their good performances because using population principle (not only one but a population of solutions is considered at each step). The genetic algorithms are nature inspired evolutionary algorithms and had been extensively used for solving the otherwise unsolvable large scale NP-hard problems. The efficiency of the genetic algorithms has been questioned owing to their stochastic nature of solution evaluations. The proposed approach presents a way around this by storing the already visited solutions in a binary space tree, which can be searched much faster as compared to other similar approaches practiced like Tabu search list.

In the algorithm developed in this paper, we eliminate the solution revisits in the context of GA. We show that an archive design using binary space partitioning (BSP) can be naturally integrated with GA so that revisits are completely eliminated (Yuen, 2009). A modification of GA by incorporating an adaptive mutation naturally arises from the integration.

3. MODEL FORMULATION

Our three stage model consists of n^i plants, n^j crossdocks and n^k markets in n^t period planning horizon. We have assumed the following

1. The units produced at the plants are routed to the markets via the crossdocks.
2. Inventory is held only at the markets.
3. Markets face a known demand for each period which is satisfied at the beginning of each period.
4. Backordering is not allowed.
5. The plants, crossdocks and markets are capacitated; their capacity does not vary with time.
6. The lead time for production and transportation of units is taken as one period each.
7. Units manufactured on each day are shipped at the beginning of the next period and no inventory costs are charged for that production period.

8. All the units take approximately the same time to reach their respective markets.
9. Deterioration has not been considered during transportation.

Figure 1 shows such a supply chain with one plant (P1), two crossdocks (CD1, CD2) and one market, with T_1, T_2, \dots, T_n denoting time periods. The plants, crossdocks and markets are capacitated; their capacity does not vary with time.

(Insert Figure 1 here)

In this model, variable lifetime of inventory has been considered. After the careful study of literature regarding the growth rate of micro-organisms Juneja and Marks (2006); Mochizuki and Hattori (1987); Martin and Felsenfeld (1963) responsible for deterioration, the deterioration function has been assumed to be exponential function. The rate of deterioration function with constants A and B is

$$Det(t) = Ae^{t/B} \quad (1)$$

The constants A and B vary with product types and the environmental conditions like seasons, under which the product is stored. The constant 'A' represents the initial deterioration rate of the product material as soon as the product leaves the production unit. The dimensional unit for 'A' is the fraction of the product material deteriorating per unit of time. The constant 'B' is the time in which the deterioration rate of the product becomes e (≈ 2.718) times of its initial value. The constant 'B' can be used to define the useful functional life for the product, after which the product should be considered as defective or unfit for consumption.

The model parameters are as follows:

α_{pt}	Fraction of units produced in p period that deteriorate in period t
C_{ij}^1	Cost of transporting a unit from plant i to crossdock j
C_{jk}^2	Cost of transporting a unit from crossdock j to market k
D_{ij}^1	Distance from plant i to crossdock j
D_{jk}^2	Distance from crossdock j to market k

The decision variables included in the model are:

N_{ijt}^{f1}	Number of <i>FTL</i> trucks shipped from plant i to crossdock j in period t
N_{ijt}^{h1}	Number of <i>HTL</i> trucks shipped from plant i to crossdock j in period t
N_{jkt}^{f2}	Number of <i>FTL</i> trucks shipped from crossdock j to market k in period t
N_{jkt}^{h2}	Number of <i>HTL</i> trucks shipped from crossdock j to market k in period t
O_{kt}	1, if order placed by market k in period t ;

		0 otherwise
d_{kt}	Demand of market k in period t	P_{it} 1, if production occurs at plant i in period t ; 0 otherwise
e_j	Handling cost per unit at the crossdock j	ptl_{ijt}^{f1} Partial full truck load shipped from plant i to crossdock j in period t
f_f	Fixed cost for hiring a single truck with full truck load (FTL)	ptl_{ijt}^{h1} Partial half truck load shipped from plant i to crossdock j in period t
f_h	Fixed cost for hiring a single truck with half truck load (HTL)	ptl_{jkt}^{f2} Partial full truck load shipped from crossdock j to market k in period t
h_k	Holding cost per unit at market k	ptl_{jkt}^{h2} Partial half truck load shipped from crossdock j to market k in period t
K_i^1	Production capacity of the plant i	V_{jkt} Units shipped from crossdock j to market k in period t
K_j^2	Capacity of the crossdock j	X_{ijt} Units shipped from plant i to crossdock j in period t
K_k^3	Capacity of the market k	Y_{kpt} Units produced in period p held as inventory at market k in period t
l	Lost cost per unit	Z_{kpt} Units produced in period p used to satisfy demand of market k in period t
n^i	Number of Plants	
n^j	Number of Crossdocks	
n^k	Number of Markets	
n^t	Number of Time periods	
s_i^1	Setup cost per production period at plant i	
s_k^2	Order cost per order at the market k	
sl	Service level	
σ_{kt}	Variance in demand of market k in period t .	

The fraction of units produced in period p that deteriorate in period t is:

$$\alpha_{pt} = \int_{t=t-p}^{t=t-p+1} Det(t)dt \quad (2)$$

The rate of deterioration of units increases with time and therefore:

$$\alpha_{p,t} > \alpha_{p+1,t} \quad (p < t) \quad (3)$$

Units produced in a specific period will deteriorate or get consumed and hence cannot increase over time:

$$Y_{kpt} > Y_{kpt^*} \quad (t^* > t) \quad (4)$$

The total holding plus lost cost for units produced in period p and held as inventory in period t becomes:

$$H(\alpha_{pt}, Y_{kpt}) = h_k(1 - \alpha_{pt})Y_{kpt} + l\alpha_{pt}Y_{kpt} \quad (5)$$

Let the number of units produced in each of the periods $p-1$ and p held as inventory at period t be equal to non-zero Q . Also, let R be non-zero and represents the number of units consumed in period $t+1$ which were produced in period p .

Property 1. If “ Z_{kpt} ” is not equal to zero, then it is sub-optimal to have “ $Y_{k,p-1,t}$ ” not equal to zero,

$$i.e., \text{ if } Z_{kpt} = Q, \text{ then } Y_{k,p-1,t} = 0 \quad (p \leq t)$$

Proof. Z_{kpt} is the number of units produced in period p used to satisfy demand of market k in period t . From equation (3), we can deduce that $H(\alpha_{p-1,t}, Q) > H(\alpha_{pb}, Q)$. Since the rate of deterioration increases with time it is more profitable to consume units from the oldest periods. Units kept as inventory in period t will be consumed in later periods. Hence, if $Y_{k,p-1,t} \neq 0$ this implies that $Z_{k,p-1,t+1} = R > 0$ and holding-plus-lost cost, $H(\alpha_{p-1,t}, R)$, will be incurred. Now, exchange $\min(Q, R)$ units between Z_{kpt} and $Z_{k,p-1,t+1}$. When Q is lesser among Q and R , then $H(\alpha_{p-1,t}, R-Q) + H(\alpha_{pb}, Q)$ is incurred which is less than $H(\alpha_{p-1,t}, R)$, ($H(\alpha_{p-1,t}, R) = H(\alpha_{p-1,t}, R-Q) + H(\alpha_{p-1,t}, Q)$). When R is less than Q , $H(\alpha_{pb}, R)$ is incurred which is again less than $H(\alpha_{p-1,t}, R)$. Thereby suggesting that an alternate solution which has $Y_{k,p-1,t} = 0$ reduces the total cost. ■

This property is very similar to the Zero inventory policy (ZIP), which has been extensively exploited to generate solutions for the classical ELS models. However, ZIP cannot be employed in the case with perishable inventory.

The transportation costs reflect economies of scale (EOS). It is a piecewise linear function which is modelled by introducing integer variables:

$$Trc(X_{ijt}) = f_i + C_{ij}X_{ijt} \quad (6)$$

where f_i is the fixed cost for hiring trucks and C_{ij} is the variable cost charged per unit.

We have considered two break-points with half truck load (HTL) and full truck load (FTL). Trucks which carry more than HTL units have been called full trucks, while others have been referred to as half trucks.

This assumption is based on the pragmatic approach followed by the various tariff agencies. Many transportation service vendors operate on less than truckload breakpoints and charge accordingly. This is increasingly getting popular especially due to the outsourcing of transportation services. Literature also shows existence of similar approaches: Vogt and Even (1972); Chu et al. (2005). However, for the sake of simplicity, we have assumed only the fixed costs for the half and full truckloads:

$$f_i = f_h \quad 0 \leq X_{ijt} \leq HTL \quad (7)$$

$$f_i = f_f \quad HTL \leq X_{ijt} \leq FTL \quad (8)$$

The safety stock is dependent on the service level (sl) and the effective variance in demand (S_{dl}). The effective variance in turn is a function of the variance in demand (S_d) and the lead time (lt). In this model the total lead time is two periods.

$$S_{dl} = S_d.lt^{1/2} \quad (9)$$

$$\text{Safety stock} = S_{dl}.sl \quad (10)$$

We minimize the total cost, the expression of which is mentioned below along with the constraints:

$$\begin{aligned} \text{Min} \{ & \sum_{t=1}^{n^t} \sum_{i=1}^{n^i} s_i^1 P_{it} \} + \sum_{t=1}^{n^t} \{ \sum_{i=1}^{n^i} \sum_{j=1}^{n^j} C_{ij}^1 D_{ij}^1 X_{ijt} + \sum_{i=1}^{n^i} \sum_{j=1}^{n^j} C_{jk}^2 D_{jk}^2 V_{jkt} \} + \sum_{t=1}^{n^t} \{ \sum_{i=1}^{n^i} \sum_{j=1}^{n^j} f_f N_{ijt}^{f1} + \sum_{i=1}^{n^i} \sum_{j=1}^{n^j} f_f N_{jkt}^{f2} + \sum_{i=1}^{n^i} \sum_{j=1}^{n^j} f_h N_{ijt}^{h1} + \sum_{i=1}^{n^i} \sum_{j=1}^{n^j} f_h N_{jkt}^{h2} \} \\ & + \sum_{t=1}^{n^t} \{ \sum_{i=1}^{n^i} \sum_{j=1}^{n^j} e_j X_{ijt} \} + \sum_{t=1}^{n^t} \sum_{k=1}^{n^k} s_k^2 O_{kt} \} + \sum_{t=1}^{n^t} \{ \sum_{k=1}^{n^k} \sum_{p=1}^t (l\alpha_{kpt} + h_k(1-\alpha_{kpt})) Y_{kpt} \} \end{aligned} \quad (P)$$

$$\sum_{j=1}^{n^j} X_{ijt} \leq K_i^1 P_{it} \quad \forall i, t \quad t = 1, \dots, n^t \quad (11)$$

$$\sum_{i=1}^{n^i} X_{ijt} \leq K_j^2 \forall j, t \quad t = 1, \dots, n^t \quad (12)$$

$$\sum_{i=1}^{n^i} X_{ijt} = \sum_{k=1}^{n^k} V_{jkt} \quad \forall j, t \quad t = 1, \dots, n^t \quad (13)$$

$$\sum_{j=1}^{n^j} V_{jkt} \leq K_k^3 O_{kt} \quad \forall k, t \quad t = 1, \dots, n^t \quad (14)$$

$$\sum_{j=1}^{nj} V_{jkt} - Z_{kt} = Y_{kt} \quad \forall k, t \quad t = 1, \dots, n^t \quad (15)$$

$$(1 - \alpha_{k,p,t-1}) Y_{k,p,t-1} - Z_{kpt} = Y_{kpt} \quad \forall k, t, p \quad 1 \leq p < t \leq n^t \quad (16)$$

$$\sum_{p=1}^t Z_{kpt} = d_{kt} \quad \forall k, t \quad t = 1, \dots, n^t \quad (17)$$

$$S_{kt} sl \leq \sum_{p=1}^t Y_{kpt} \leq K_k^3 \quad \forall k, t \quad t = 1, \dots, n^t \quad (18)$$

$$N_{ijt}^{f1} = \left\lfloor \frac{X_{ijt}}{FTL} \right\rfloor \quad \forall i, j, t \quad t = 1, \dots, n^t \quad (19)$$

$$N_{jkt}^{f2} = \left\lfloor \frac{V_{jkt}}{FTL} \right\rfloor \quad \forall j, k, t \quad t = 1, \dots, n^t \quad (20)$$

$$N_{ijt}^{h1} = \left\lceil \frac{X_{ijt} - N_{ijt}^{f1} \cdot FTL}{HTL} \right\rceil \quad \forall i, j, t \quad t = 1, \dots, n^t \quad (21)$$

$$N_{jkt}^{h2} = \left\lceil \frac{V_{jkt} - N_{jkt}^{f2} \cdot FTL}{HTL} \right\rceil \quad \forall j, k, t \quad t = 1, \dots, n^t \quad (22)$$

$\lfloor x \rfloor$ is the greatest integer function and $\lceil x \rceil$ is the least integer function. Besides above constraints, the non-negativity constraints for the variables have been assumed implicitly.

The function (P) to be minimized, the expected total cost ($[Tc]$), includes the manufacturer and buyer costs. The manufacturer cost consists of the setup costs at the plants, the variable and fixed transportation costs and handling costs incurred at the crossdocks. The buyer cost consists of the setup costs per order, holding costs and the lost cost of deteriorated units. The inequalities (11) and (12) are the capacity constraints for the plants and the crossdocks. The outflow of units through the crossdocks should be equal to the inflow (Constraint (13)). The inequality (14) binds the binary variable of order cost to the units reaching each market. Since in case the retailer does not put an order (i.e. $O_{kt} = 0$) then no units should be shipped to that market, in that particular period (i.e. $V_{jkt} = 0$). Equation (15) states that after consuming Z_{kt}

units from the lot V_{jkt} to satisfy demand, the remaining units are kept as inventory. Equation (16) balances inventory across consecutive periods after taking into account the deterioration in each period. Since backlogging is not allowed the demand must be met by units produced in earlier or the current period (Constraint (17)). For each market, the inventory in any period cannot exceed its maximum capacity and each market has to maintain a minimum level of inventory as safety stock (Constraint (18)). The number of full trucks is a multiple of FTL for transportation from plants to crossdocks and from crossdocks to markets (Constraint (19), (20)). The remaining truck load is sent by half trucks (Constraint (21), (22)).

3.1 NON-REVISITING GENETIC ALGORITHM (NRGA)

NrGA is a variation of evolutionary (or genetic) algorithms, motivated by the complete elimination of revisits by storing already visited solutions in an archive design using binary space partitioning (BSP). The flow of the algorithm is demonstrated using a flow chart diagram in Figure 2.

3.1.1 Chromosome Encoding and Evaluation

One of the ways to represent the solution by a chromosome is to directly use the values of the variables as alleles. However, if the variables differ by large values then it is not possible to represent them using a common set of alleles. We have overcome this limitation by first normalizing the variables using the upper and lower bounds for their values as 1 and 0 respectively. These normalized discrete values are then stored in the genes of the chromosome. We call the discrete interval size as resolution limit, which depends on the availability of computation power. Smaller is the resolution limit, more refined is the search and thus, better is the solution. The decoding of the chromosome requires calculating the real values of the variables from the normalized values stored in the genes of the chromosome. It should also be noted that the conventional encoding schemes for transportation problems, like priority encoding scheme, can hinder the functionality of the parameter-less mutation operator, which is an important characteristic of the proposed algorithm.

(Insert Figure 2 here)

3.1.2 Population Initialization, Selection and Crossover

NrGA is a population based optimization strategy. The initial population is generated randomly using the pseudo-random numbers. A popular method for selection in GA is Roulette Wheel Selection used in our algorithm. The chromosome selection probability is inversely proportional to its fitness value. We choose single point crossover where the reference point is decided randomly over chromosome length.

3.1.3 Dynamic Tree Archive

NrGA may be summarized as a GA that interacts with a BSP tree archive (Ar), which stores the solutions visited at least once and its fitness value. Immediately after the crossover, the archive stores the solutions based on the values of the dimensions of the solution space and not based on the fitness value of the individuals. However, it compares the fitness value of the candidate individual with those of the existing ones in the tree and reports revisit if true as shown in Figure 2. In case of revisits, the archive doesn't reject the solution, but tries to obtain another solution from the revisited solution. The archive employs a self-adaptive mutation operator on the revisited solution to obtain a new solution, and then repeats the cycle for candidate insertion in BSP tree. As a result, the number of BSP tree nodes is equal to the number of generated GA solutions.

Since, the implementation of NrGA involves the representation of the solution space through the archive, the BSP tree nodes are supposed to represent the sections of the solution space. Hence, it is required that the parent node represents the solution space section which is the union of the disjoint solution space sections of its children nodes at any given level, below the level of the parent node, in the tree. Thus, it can be concluded that the root of the tree represents the entire solution space, and each child node represents the subspace of the solution search space of the parent node. However, in case of multi-dimensional search space, a slight aberration is required to incorporate the location of the solution stored in the node, into the search subspace associated with the node.

(Insert Figure 3, 4 and 5 here)

Let's consider an example to demonstrate the working of NrGA. Let the objective function (S) consists of

three variables, viz. x , y and z . Thus, the solution search space has three dimensions and each dimension may or may not have a unique resolution limit. Each BSP tree node consists of objective function value of the solution it stores and the search space it represents in the BSP. Let the possible values of the variables be as follows:

$$x = \{1, 2, 3, 4, 5\} \quad y = \{2, 4, 6, 8\} \quad z = \{5, 7, 11, 13\}$$

The number of maximum possible distinct solutions is, $4 \times 4 \times 5 = 80$, which itself is a function of the resolution limits, and the upper and lower bounds of the variables. Depending on the availability of the computation power and the accuracy of the search, one can set these limits accordingly.

Let $S1 = \{2, 6, 11\}$ be the first solution generated by GA. The search space associated with it is the entire solution search space and it is inserted in the root node of the archive tree. Note that the solution stored in any node must lay in the search subspace represents by the same node (Figure. 3). The second iteration generates $S2 = \{3, 4, 7\}$. Preferring left node over right node, $S2$ is stored in left node. In the absence of second child node, the solution space of the existing child node is same as that of the parent node (Figure 4.) except for the solution $\{2, 6, 11\}$.

The third iteration generates $S3 = \{1, 8, 5\}$. Since the right node is empty, it is stored in the right node and shares the search space with left node. In order to define new subspaces for the two child nodes, the point of separation for each dimension, like x , is calculated by taking the mean of the values of the dimension in the two solutions, e.g. the mean of the value of x in $S2$ and in $S3$, i.e., $(3+1)/2 = 2$, and again the tie is broken by giving priority to the left part of dimension space for inclusion of the point of separation in search subspace. Thus the search space for dimension x for $S3$ becomes $\{1, 2\}$ and for $S2$ becomes $\{3, 4, 5\}$, see Figure 5. $S3$ is the right node, if the priority is for the left one so it is necessary to move 2 to $S2$. In case of non-integer separation point, values of dimension to left and right of separation point are allotted similarly.

Further iteration generated $S4 = \{5, 2, 11\}$. When a parent node has two child nodes and a new node is required to be inserted then a comparison is carried out between the left and the right node for the selection

of the node to append the new node. First, all the variables are normalized using the following relation,

$$\text{normalized value} = \frac{(\text{value} - \text{lower bound})}{(\text{upper bound} - \text{lower bound})}$$

The normalization of the variables is necessary to preclude the effect of their magnitudes. Then the difference between the normalized values of both the nodes for each dimension is calculated, and the dimension with the highest difference is selected. Now for the selected dimension, the difference between the normalized values of the variables of the node to be inserted and the two child nodes is calculated. The new node is added to the child node having the lesser difference with it. When $S4$ is inserted, the values are,

$$S2_{nx} = (3 - 1) / (5 - 1) = 0.50$$

$$S2_{ny} = (4 - 2) / (8 - 2) = 0.33$$

$$S2_{nz} = (7 - 5) / (13 - 5) = 0.25$$

$$S3_{nx} = (1 - 1) / (5 - 1) = 0.00$$

$$S3_{ny} = (8 - 2) / (8 - 2) = 1.00$$

$$S3_{nz} = (5 - 5) / (13 - 5) = 0.00$$

The difference is calculated as follows:

$$\begin{aligned} |S2 - S3| &= \{|S2_{nx} - S3_{nx}|, |S2_{ny} - S3_{ny}|, |S2_{nz} - S3_{nz}|\} \\ &= \{0.5, 0.67, 0.25\}. \end{aligned}$$

The difference is greatest for dimension y , hence it is selected. Normalized value of y dimension for $S4$ is:

$$S4_{ny} = (2 - 2) / (8 - 2) = 0.0$$

$$\text{Difference with } S2: |S4_{ny} - S2_{ny}| = 0.33$$

$$\text{Difference with } S3: |S4_{ny} - S3_{ny}| = 1.0$$

The difference of the y dimension of $S4$ from $S2$ is lesser than the difference with $S3$. Therefore, the new solution $S4$ is added to the node containing the solution $S2$.

(Insert Figure 6 and 7 here)

Resulting BSP tree after the addition of solutions $S5 = \{4, 4, 5\}$, $S6 = \{3, 8, 13\}$ and $S7 = \{5, 6, 7\}$ is shown in Figure 6. Arrangement of all the seven solutions within the solution search space is shown in Figure 7.

For a problem with large number of variables like ours, it is highly unlikely to visit the same location in search space again. An alternative way to report a subspace as a singleton set is to compare the value of all the variables of the solution and the subspace, and report it as a singleton subspace if 90 percent (say) of the variables have same value for solution and subspace. This is similar to the aspiration criterion used in Tabu Search and referred as singleton claim limit (SCL) in our algorithm. The SCL should be kept high, ideally 100 percent, for refined search of solution space. Thus, resolution limits and SCL are the defining parameters for the accuracy of the solution provided by our NrGA.

3.1.4 Self-Adaptive Mutation Operator

The point of departure between NrGA and traditional GAs is the self-adaptive mutation operator employed in NrGA, unlike a fixed mutation rate in conventional GAs. Soon after the algorithm realises that the solution offered by GA is already present in the archive tree, it randomly mutates the chromosome to another solution within the subspace associated with the existing solution. Note that feasibility criterion for the mutated solution is that it must lie within the subspace of the considered existing solution. If there is no space for another solution, i.e., either the subspace is a singleton set or the entire search subspace has been visited, then the algorithm backtracks the subspace of the immediate parent and uses it for the feasibility criterion, and so on. Our mutation operator is adaptive in the sense that the new solution obtained after mutation is a function of both original solution and the unexplored region in the solution search space. The step size of the mutation itself is a function of its position in the archive tree. The more visited is a particular branch of the archive tree, more is the exploration of search space along that branch, hence less in the unvisited space available for the mutated solution. Thus, the step size of mutation is small for frequently visited subspace, resulting in a more refined search. Similarly, for large unvisited solution space, mutation step size is large.

3.2 Results

The proposed model and the NrGA algorithm were programmed in Sun Microsystems' JAVA, JDK (Java Development Kit) 1.6. All code was executed on a personal computer with an Intel® Core2 Duo CPU T8100 @ 2.10GHz, 2.10GHz processor and 3.00GB RAM (Random Access Memory) on a 32-bit operating system.

As the proposed model is first of its kind for the integration of supply chain for perishable products with truckload discounts, there is no commonly used benchmark model for the perishable products. Yang et al. (2000) randomly generated the locations of the depots and the customers. Bianchi et al. (2004) highlighted that there is no commonly used benchmark for the Vehicle Routing Problem with Stochastic Demand (VRPSD) in the literature. The data used for our problem have been synthetically generated based on the careful study of the, earlier addressed, similar problems in the literature. The data used for our problem is listed in Table 1.

(Insert Table 1 here)

3.2.1 Computational Results and Analysis

We established a favorable choice of parameters by means of systematic experimentation. Since the algorithm is stochastic in nature, its performance is evaluated based on statistics obtained from 50 independent runs and all experimental results were averaged over 50 trials. We carried out simulations using favorable combination of genetic parameters, obtained by parameter tuning for NrGA, as listed in Table 2.

(Insert Table 2 and 3 here)

We show and compare the results in the graphical format, and list the results of 10 trials in Table 3. Figure 8 demonstrates the convergence of NrGA for our three-stage supply chain model with perishable products and truckload discounts. It also shows the comparison of NrGA with traditional genetic algorithm for the same model settings. Since fitness value of the chromosome is proportional to the total cost for the

model, the convergence of fitness function also assures the convergence of costs for our model.

(Insert Figure 8 and 9 here)

Production cost indicated in Figure 9 includes the ordering cost and handling costs used in our model. It is clear from the figure that the transportation costs constitute about 70 % of total cost, followed by production costs for the products to be about 29 %. The holding and lost costs for the perishable products constitute less than 1 % of the total cost. The total cost will increase with time because of the lost cost incurred for the products perished in the subsequent periods. It is expected that the total cost for each period, for a set of stable demand, will later stabilize with time.

Figure 10 shows the inventory levels at different periods of time by both considering and neglecting the perishable nature of the products for our model. The inventory level of the solution provided by our algorithm and model is slightly higher than others. This can be justified by the fact that the contribution of the inventory holding cost towards the total cost is negligible (less than 1%). The inventory peak at period 3 is because of the lower demand of period 3 as compared to period 2 and period 3. Period 5 has the least demand among all the 5 periods and is preceded by the high demand at period 4, which explains the high inventory level and holding costs for period 5 as compared to all other periods.

(Insert Figure 10 and 11 here)

The holding and lost cost for our model-algorithm combination is shown in Figure 11. Please note that the inventory levels shown in Figure 10 are at the beginning of the periods whereas the holding costs in Figure 11 are calculated towards the end of the periods. The lost cost per unit is significantly higher than the holding cost per unit of product. The higher lost cost incurred in successive periods is due to the exponentially increasing deterioration rate of the product with time. The savings on lost cost depends on the ability of the firm to decrease the deterioration rate of products, either by altering the chemical composition, adding preservatives or better storage facilities.

The production cost, ordering cost and handling cost, all taken together have been plotted in Figure 12 as production costs. The costs for non-perishable products have been calculated from our model by setting the

value of deterioration rate to zero. The plot shows that a reduction of about 5 percent of total supply chain cost is achieved by considering the perishable nature of the products during policy making for supply chain.

The transportation costs constitute about 70 percent of the total supply chain costs. One of the strategies to reduce transportation costs and also incorporated in our model is to provide truckload discounts. Instead of using trucks with two different capacities viz. FTL and HTL, we used trucks with only one capacity $(HTL + FTL) / 2$. The choice of the capacity $(HTL + FTL)/2$ for no truckload discounts is based on the pragmatic scenarios where the transportation service vendors prefer to use the medium-sized vehicles for no truckload discounts, and large-capacity vehicles for truckloads with discounts. Figure 13 describes the comparison of transportation costs between two cases, i.e. with and without truckload discounts. About 20 percent reduction in total supply chain costs is achieved by the introduction of truckload discounts.

(Insert Figure 12 and 13 here)

Based on the aforementioned features of our model, the sensitivity analysis can be carried out for various model parameters. For example, when fixed transportation costs, i.e., f_f and f_h , for hiring a truck are increased, then a sharp decline is observed in the ordering costs. This arises due to the EOS factor; as the cost of transportation increases it becomes cheaper to ship goods less number of times and hence the reduction in the number of orders. The holding costs increases with replenishment period. It was also observed that the variance was symmetric about the base case. This was observed for each of the cases except for FTL. We noticed that total cost does not vary much with each of the parameters. Except for FTL (3%) and total demand (2.3%), the variation in total cost was less than 1.5%.

3.2.2 Model Robustness under Varied Demand Scenario

This section shows that the ability of our model to operate on randomly generated demand sets can lead to solutions which are more robust to the stochastic nature of the problem compared to the deterministic approach and that the expected costs of such solutions are good estimates of their true performance.

We check our model for a varied demand scenario. The demand generated is normally distributed. For this analysis we do not include the costs incurred due to unsatisfied demand. We performed calculations by

varying cost parameters and find the ratio of the average Total cost (Γ , averaged over 50 instances) to the expected total cost ($[Tc]$), and the standard deviation (Δ) in the average total cost (Table IV). $\Gamma/[Tc]$ is very close to 1 and we expect that by calculating $\Gamma/[Tc]$ for a large sample the ratio would converge to 1. We can observe that Δ is insensitive to S_{kt} (variance in demand). This showcases the robustness of our formulation due to its applicability to changing demand patterns.

(Insert Table IV here)

3.2.3 Fleet selection

In real world scenario, freight transport companies provide a variety of carriers with different capacities. We have considered a cost structure that reflects economies of scale, for such carriers. The analysis carried out for our model, by varying the demand to higher levels and lower levels, shows that it becomes relatively cheaper to hire trucks with larger capacity as demand increases, see Table 5. It clearly shows that hiring a fleet with FTL of 250 is slightly cheaper than hiring one with FTL of 150. It is expected that as the demand increases the minima would shift towards fleets with higher capacities. We tested for a different, higher set of demand values and found the results as expected. Table 6 shows that a fleet with 450 FTL would serve the best for this particular scenario. The reason is that for FTL values higher than 450 the total partial truck loads shipped are higher due to which more trucks have to be hired, while for FTL values less than 450, EOS plays into effect; resulting, for both the cases, in an increase in the total transportation cost.

(Insert Table 5 and 6 here)

3.2.4 Centralized and Decentralized Supply Chain

For a decentralized chain, we examine the cost borne by the vendor and the buyer and compare it with those for the centralized chain. For this, we alter the FTL values while keeping all other parameters constant. We assume that the decentralized chains would be operated from the point of view (POV) of the buyer.

(Insert Figure 14, 15 and 16 here)

Please note that in Figure 14, Figure 15, and Figure 16, Integrated refers to Integrated Supply Chain, Vendor refers to VPOVC (Vendor Point of View Chain) and Buyer refers to BPOVC (Buyer Point of View Chain).

From Figure 14, we can interpret that for different FTLs, the cost sharing between the vendor and the buyer differ significantly. For supply chains, where vendor cost is minimized, we call these 'the vendor POV chain (VPOVC)', we see that as the truck load increases the total cost reduces which implies that the cost reductions by EOS are significantly larger than the additional holding costs incurred by the buyer. A similar trend is observed for the centralized/integrated chain (IC) for the same reason. However, when the buyer cost is minimized (denoted by 'the buyer POV chain (BPOVC)'), the total cost follows a different pattern with a minima at FTL equal to 350. We can also observe that the points for FTL 350 coincide for BPOVC and IC. The vendor would gain most from this scenario. While in the other cases, he/she would have to offer substantial discounts to the buyer to move from BPOVC to IC. The buyer cost would essentially remain constant over varied FTL since holding cost is not affected by truck capacity and therefore, the difference between BPOVC line and the IC line would represent the vendor cost reduction. For a move from BPOVC to IC appropriate discounts will have to be given to the buyer. Figure 15 shows the cost reductions for the vendor and the buyer for such a move and any increase in buyer cost will have to be discounted for, by the vendor.

We return to the problem of fleet selection and analyze it from the vendor's perspective. For decentralized supply chain with FTL equal to 250, we can observe maxima for both VPOVC and BPOVC, as shown in Figure 16. Although, the same point has a minima for the centralized supply chain; it is the operating point for the centralized chain. The additional costs, i.e. the difference between the centralized and decentralized supply chain costs, incurred are traded between the vendor and the buyer moving from VPOVC to BPOVC. In the centralized supply chain these additional costs are balanced between them. The most suitable options for the vendor are FTL equal to 150 and FTL equal to 350 when BPOVC is considered.

(Insert Table VII here)

For the vendor a move to *FTL* equal to 250 is only encouraged when decision making is centralized. This figure shows that fleet selection decisions depend on the structure of the supply chain as well. The total cost is proportionate to the lead time because of deterioration while shipping of units and safety stock (Table 7). If the cost of changing the mode of transportation is covered by the profit made due to the reduction in lead time; the change of mode is accepted.

3.2.5 *Algorithm Convergence and Performance*

To evaluate the impact of the proposed non-revisiting algorithm, we compare the optimal fitness found by NrGA with a traditional GA. Premature convergence is an undesirable phenomenon often reported in the literature as to cause poor performance of GA. The plots in Figure 17 clearly demonstrate the ability of NrGA to avoid premature convergence. To be a practical solution to real problem, the processing time of NrGA should be within a reasonable range. Thus, the overhead for NrGA related to traditional GA is also observed. The average computational times for NrGA and GA are compared in Figure 18.

(Insert Figure 17 and 18 here)

For benchmarking, the proposed approach has been compared with the solution obtained from IBM's ILOG CPLEX 12.1 software. For this, the problem constraints were modified to formulate the problem as Mixed Integer Problem for CPLEX (See Appendix A).

(Insert Table 8 here)

The results have been obtained for small sized problems and presented in Table 8. The CPLEX solver uses a mixed integer programming model and solves the problem using branch-and-bound cut method. The computational experiments show that the proposed algorithm is competitive, finding near optimal solutions in most instances, demanding short computing times.

4 CONCLUSION

This paper presents a modified genetic algorithm for solving the proposed production-inventory-distribution (PID) model for multi-stage supply chain with perishable products and truckload discounts. To the best of our knowledge, it is the first time that a non-revisiting genetic algorithm with parameter-less self-adaptive mutation operator has been used to solve this kind of problem. In fact, the proposed model itself refers to the issues that have not been investigated thoroughly in the literature and are of increasing relevance to the cosmetics, FMCG (Fast Moving Consumer Goods), pharmaceutical and other similar industries. We show that by considering the perishable nature of the products while determining the optimum policy for supply chain, significant cost reductions have been realised. Further coupling it with existing cost-cutting strategies like truckload discounts has shown tremendous improvements in the results. The beauty of our model lies in capturing the deterioration nature of the products, and the integration across the multi-stage supply chain.

Extensive simulations have been performed to show that the solutions obtained by our model, equipped with the non-revisiting GA, are robust to the varied nature of the problem. We also demonstrate how our model shows the significance of economies of scale (EOS) in transportation, and discuss the related problem of fleet selection and optimal vehicle capacity. Fleet selection could be considered to be dependent on two parameters, fixed cost per truck and lead time associated with it. There is no mathematical formulation to decide whether crossdocking would be more suitable or warehousing (see Appendix B). However, a study on the structure of the chain keeping in mind certain factors can aid in making this decision.

The results for the proposed model have been obtained by the non-revisiting GA, and the conventional GA. The former has been found to perform better than the latter algorithm. The non-revisiting GA was tested using a set of data problems and the results were validated by running the CPLEX optimizer with the same data. This solver used a mixed integer programming model also developed in this work. The computational experiments show that the non-revisiting GA is competitive, finding near optimal solutions

in most instances, demanding short computing times.

APPENDIX A

Formulating the Model as a MIP

The greatest and least integer functions in constraints (19) to (22) make the problem non-linear. These constraints are replaced by another set of constraints to reduce the model to a mixed integer problem. For transportation from plant i to crossdock j in period t , the maximum possible number of units are shipped by N^{f1}_{ijt} full trucks (Constraints (23), (24)), while the remaining units (ptl^{f1}_{ijt}) are shipped by the required number of half trucks, N^{h1}_{ijt} (Constraints (25), (26)).

$$N^{f1}_{ijt} FTL + ptl^{f1}_{ijt} = X_{ijt} \quad \forall i, j, t \quad (23)$$

$$ptl^{f1}_{ijt} < FTL \quad \forall i, j, t \quad (24)$$

$$(N^{h1}_{ijt} - 1)HTL + ptl^{h1}_{ijt} = ptl^{f1}_{ijt} \quad \forall i, j, t \quad (25)$$

$$ptl^{h1}_{ijt} < HTL \quad \forall i, j, t \quad (26)$$

For transportation from crossdock j to market k in period t , similar constraints can be written to remove non-linearity from equations (20) and (22).

$$N^{f2}_{jkt} FTL + ptl^{f2}_{jkt} = V_{jkt} \quad \forall j, k, t \quad (27)$$

$$ptl^{f2}_{jkt} < FTL \quad \forall j, k, t \quad (28)$$

$$(N^{h2}_{jkt} - 1)HTL + ptl^{h2}_{jkt} = ptl^{f2}_{jkt} \quad \forall j, k, t \quad (29)$$

$$ptl^{h2}_{jkt} < HTL \quad \forall j, k, t \quad (30)$$

The MIP problem was solved using the IBM's ILOG CPLEX 12.1 software, which used the branch-and-bound/cut method. The code for this MIP was written in Java using the ILOG Concert Technology developed for CPLEX.

REFERENCES

- [1] Aggarwal, A., & Park, J. K. (1993). Improved algorithms for economic lot size problems. *Operations Research*, 41(3), 549-571.
- [2] Ai, X. Y., Zhang, J. L., & Wang, L. (2017). Optimal joint replenishment policy for multiple non-instantaneous deteriorating items. *International Journal of Production Research*, 55(16), 4625-4642.
- [3] Aloulou, M.A., Dolgui, A. & Kovalyov, M.Y. (2014). A bibliography of non-deterministic lot-sizing models, *International Journal of Production Research*, 52(8), 2293–2310.
- [4] Alfares, H. K., Khurshed, S. N., & Noman, S. M. (2005). Integrating quality and maintenance decisions in a production-inventory model for deteriorating items. *International Journal of Production Research*, 43(5), 899-911.
- [5] Bai, R., Burke, E. K., & Kendall, G. (2008). Heuristic, meta-heuristic and hyper-heuristic approaches for fresh produce inventory control and shelf space allocation. *Journal of the Operational Research Society*, 59(10), 1387-1397.
- [6] Boudia M., Prins C., (2009). A memetic algorithm with dynamic population management for an integrated production-distribution problem, *European Journal of Operational Research*, 195 (3), 703-715.
- [7] Buijs, P., Vis, I.F.A. & Carlo, H.J., (2014). Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research*, 239(3), 593-608.
- [8] Benton, W. C., & Park, S. (1996). A classification of literature on determining the lot size under quantity discounts. *European Journal of Operational Research*, 92(2), 219-238.
- [9] Bianchi, L., Birattari, M., Chiarandini, M., Manfrin, M., Mastrolilli, M., Paquete, L., ... & Schiavinotto, T. (2004). Metaheuristics for the vehicle routing problem with stochastic demands. In: Proceedings of the International Conference on Parallel Problem Solving from Nature (pp. 450-460). Springer: Berlin Heidelberg.

- [10] Brandão, J. (2009). A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 195(3), 716-728.
- [11] Chandra, P., & Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3), 503-517.
- [12] Chare, P., & Schrader, G. (1963). A model for exponentially decaying inventories. *Journal of Industrial Engineering*, 15, 238-243.
- [13] Chen, P., Guo, Y., Lim, A., & Rodrigues, B. (2006). Multiple crossdocks with inventory and time windows. *Computers & Operations Research*, 33(1), 43-63.
- [14] Choi, E., & Tcha, D. W. (2007). A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34(7), 2080-2095.
- [15] Chu, L. Y., Hsu, V. N., & Shen, Z. J. M. (2005). An economic lot-sizing problem with perishable inventory and economies of scale costs: Approximation solutions and worst case analysis. *Naval Research Logistics*, 52(6), 536-548.
- [16] Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568-581.
- [17] Dolgui, A. & Proth J.-M., (2010) *Supply chain engineering: Useful methods and techniques*, Springer: London.
- [18] Erengüç, Ş. S., Simpson, N. C., & Vakharia, A. J. (1999). Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, 115(2), 219-236.
- [19] Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2), 109-124.
- [20] Fries, B. E. (1975). Optimal ordering policy for a perishable commodity with fixed lifetime. *Operations Research*, 23(1), 46-61.

- [21] Fu, L. L., Aloulou, M. A., & Triki, C. (2017). Integrated production scheduling and vehicle routing problem with job splitting and delivery time windows. *International Journal of Production Research*, 1-16.
- [22] Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2), 340-349.
- [23] Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1), 49-66.
- [24] Hauer, E. (1971). Fleet selection for public transportation routes. *Transportation Science*, 5(1), 1-21.
- [25] Hsu, V. N. (2000). Dynamic economic lot size model with perishable inventory. *Management Science*, 46(8), 1159-1169.
- [26] Jiang, H., & Yang, J. (2014). A Multi-attribute Classification Method on Fresh Agricultural Products. *Journal of Computers*, 9(10), 2443-2448.
- [27] Juneja, V. K., & Marks, H. M. (2006). Growth kinetics of Salmonella spp. pre-and post-thermal treatment. *International Journal of Food Microbiology*, 109(1), 54-59.
- [28] Kaminsky, P., & Simchi-Levi, D. (2003). Production and distribution lot sizing in a two stage supply chain. *IIE Transactions*, 35(11), 1065-1075.
- [29] Kratica, J. (1999). Improving performances of the genetic algorithm by caching. *Computers and Artificial Intelligence*, 18(3), 271-283.
- [30] Langevin, A., & Riopel, D. (Eds.). (2005). *Logistics systems: design and optimization*. Springer Science & Business Media.
- [31] Li, C. L., Hsu, V. N., & Xiao, W. Q. (2004). Dynamic lot sizing with batch ordering and truckload discounts. *Operations Research*, 52(4), 639-654.
- [32] Lima, C. D. R., Goldberg, M. C., & Goldberg, E. F. G. (2004). A memetic algorithm for the heterogeneous fleet vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 18, 171-176.

- [33]Lin, C. W. R., & Chen, H. Y. S. (2003). Dynamic allocation of uncertain supply for the perishable commodity supply chain. *International Journal of Production Research*, 41(13), 3119-3138.
- [34]Lütke Entrup, M., Günther, H. O., Van Beek, P., Grunow, M., & Seiler, T. (2005). Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research*, 43(23), 5071-5100.
- [35] Martin, R. G., & Felsenfeld, G. (1964). A new device for controlling the growth rate of microorganisms: The exponential gradient generator. *Analytical Biochemistry*, 8(1), 43-53.
- [36] Mochizuki, M., & Hattori, T. (1987). Kinetic study of growth throughout the lag phase and the exponential phase of *Escherichia coli*. *FEMS Microbiology Ecology*, 3(5), 291-296.
- [37] Nahmias, S. (1982). Perishable inventory theory: A review. *Operations Research*, 30(4), 680-708.
- [38]Nakandala, D., Lau, H., & Shum, P. K. (2017). A lateral transshipment model for perishable inventory management. *International Journal of Production Research*, 1-14.
- [39]Napolitano, M. (2002). *Making the move to crossdocking—a practical guide*. Warehousing Education and Research Council.
- [40]Ochi, L. S., Vianna, D. S., Drummond, L. M., & Victor, A. (1998). A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. *Future Generation Computer Systems*, 14(5-6), 285-292.
- [41]Povinelli, R. J., & Feng, X. (1999). Improving genetic algorithms performance by hashing fitness values. In *Proceedings of the International Conference on Artificial Neural Networks in Engineering*, World Academy of Science, Engineering and Technology, (pp. 399-404).
- [42]Ronald, S. (1998). Duplicate genotypes in a genetic algorithm. In *Evolutionary Computation Proceedings, 1998, IEEE World Congress on Computational Intelligence: 1998 IEEE International Conference on Evolutionary Computation*, (pp. 793-798). IEEE.
- [43]Schaffer, B. (2000). Implementing a successful crossdocking operation. *PLANT ENGINEERING-CHICAGO THEN HIGHLANDS RANCH*, 54(3), 128-135.

- [44] Tarantilis, C. D., & Kiranoudis, C. T. (2001). A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of Food Engineering*, 50(1), 1-9.
- [45] Thomas, D. J., & Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research*, 94(1), 1-15.
- [46] Van Hoesel, S., Romeijn, H. E., Morales, D. R., & Wagelmans, A. P. (2005). Integrated lot sizing in serial supply chains with production capacities. *Management Science*, 51(11), 1706-1719.
- [47] Vogt, L., & Even, J. (1972). Piecewise linear programming solutions of transportation costs as obtained from rate tariffs. *AIIE Transactions*, 4(2), 148-153.
- [48] Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science*, 5(1), 89-96.
- [49] Wang, X., Li, D., & O'Brien, C. (2009). Optimisation of traceability and operations planning: an integrated model for perishable food production. *International Journal of Production Research*, 47(11), 2865-2886.
- [50] Wang, X., Wu, Y., Liang, L., & Huang, Z. (2016). Service outsourcing and disaster response methods in a relief supply chain. *Annals of Operations Research*, 240(2).
- [51] Wassan, N. A., & Osman, I. H. (2002). Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*, 53(7), 768-782.
- [52] Wolsey, L. A. (1995). Progress with single-item lot-sizing. *European Journal of Operational Research*, 86(3), 395-401.
- [53] Yang, W. H., Mathur, K., & Ballou, R. H. (2000). Stochastic vehicle routing problem with restocking. *Transportation Science*, 34(1), 99-112.
- [54] Yousefikhoshbakht, M., Didehvar, F., & Rahmati, F. (2014). Solving the heterogeneous fixed fleet open vehicle routing problem by a combined metaheuristic algorithm. *International Journal of Production Research*, 52(9), 2565-2575.

[55]Yuen, S. Y., & Chow, C. K. (2009). A genetic algorithm that adaptively mutates and never revisits. *IEEE Transactions on Evolutionary Computation*, 13(2), 454-472.

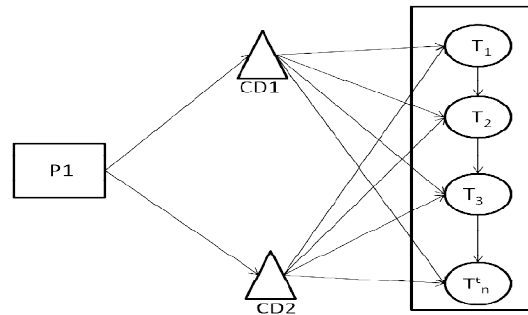


Figure 1. Two stage model with 1 plant, 2 crossdocks and 1 market with n^t time periods

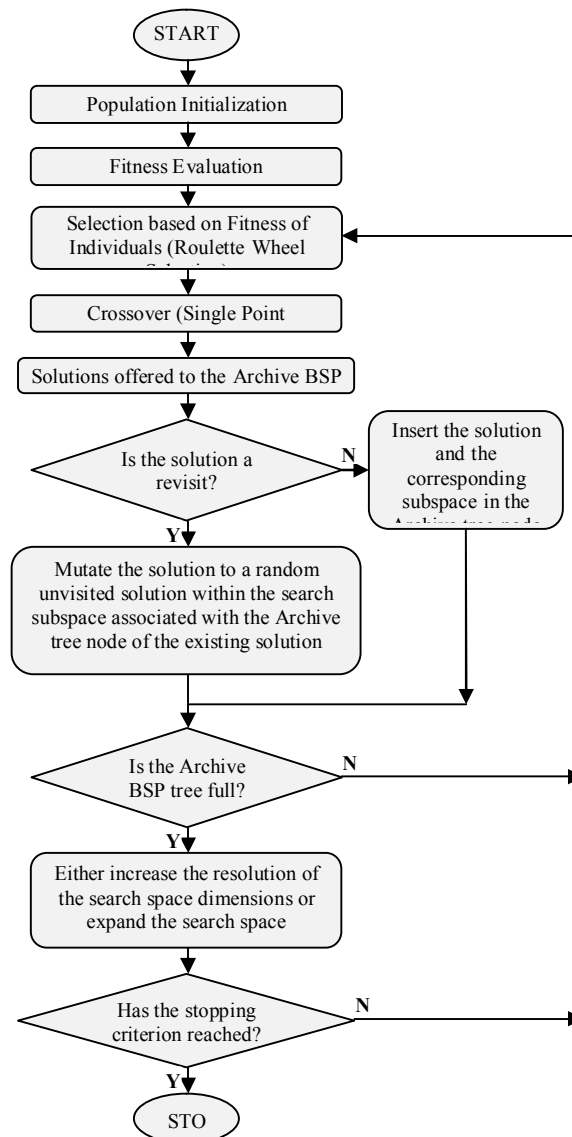


Figure 2. Flowchart of NrGA

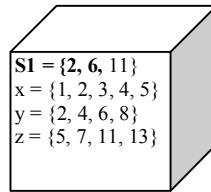


Figure 3. Root Node

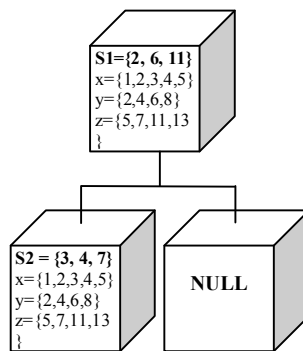


Figure 4. Tree with S1 and S2

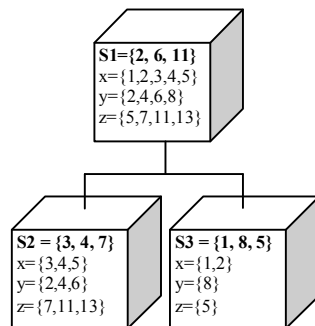


Figure 5. Tree with S1, S2 and S3

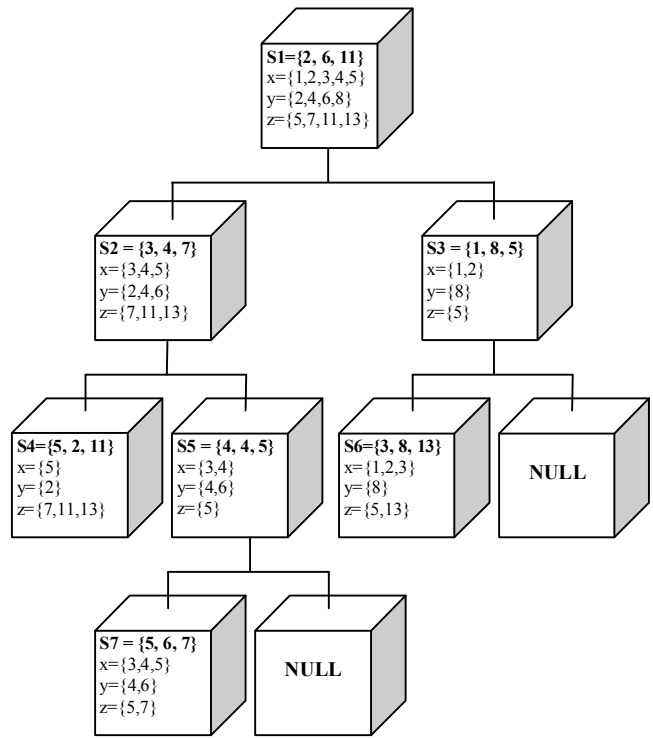


Figure 6. Binary Space Partition (BSP) Tree with all the seven solutions

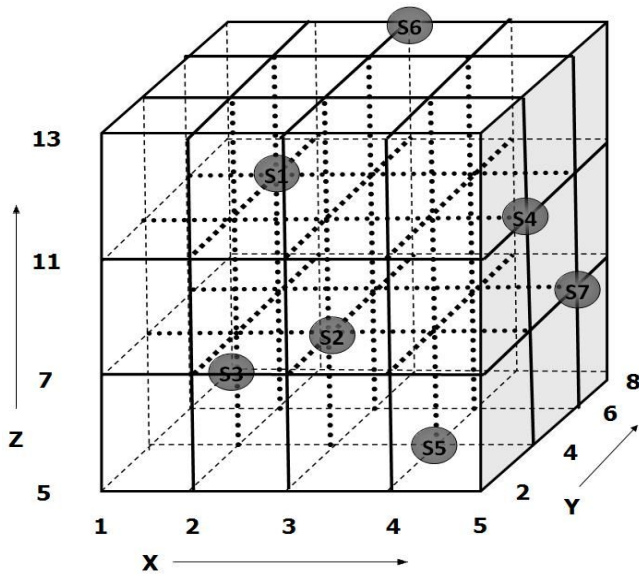


Figure 7. Solutions located in search space (root node)

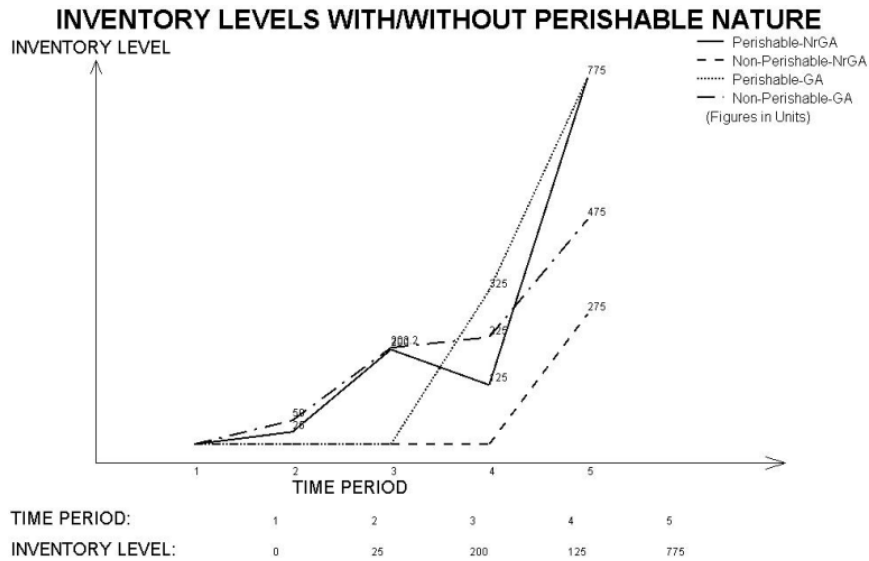


Figure 10. Inventory at the beginning of each period

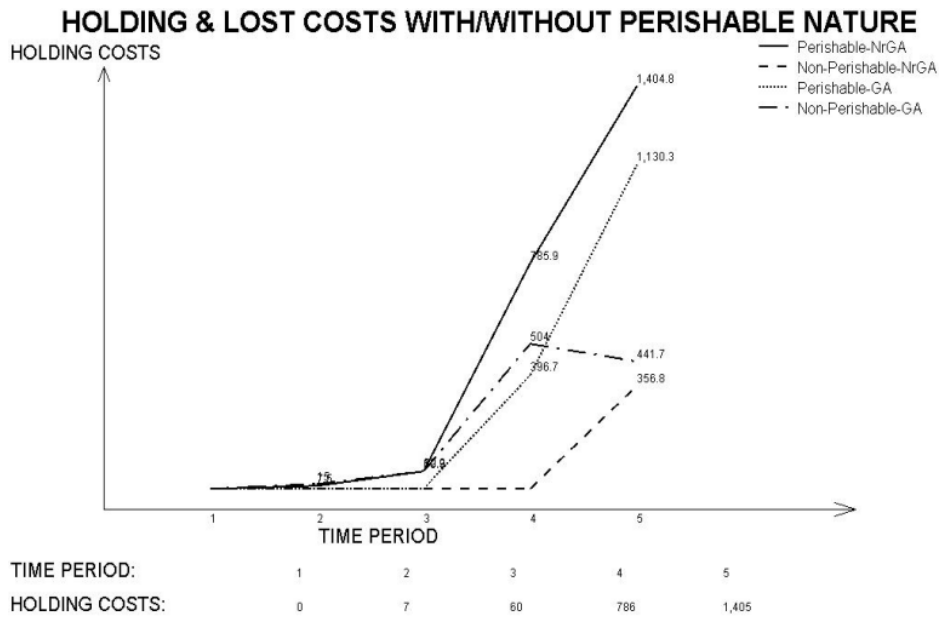


Figure 11. Holding and Lost costs for each period

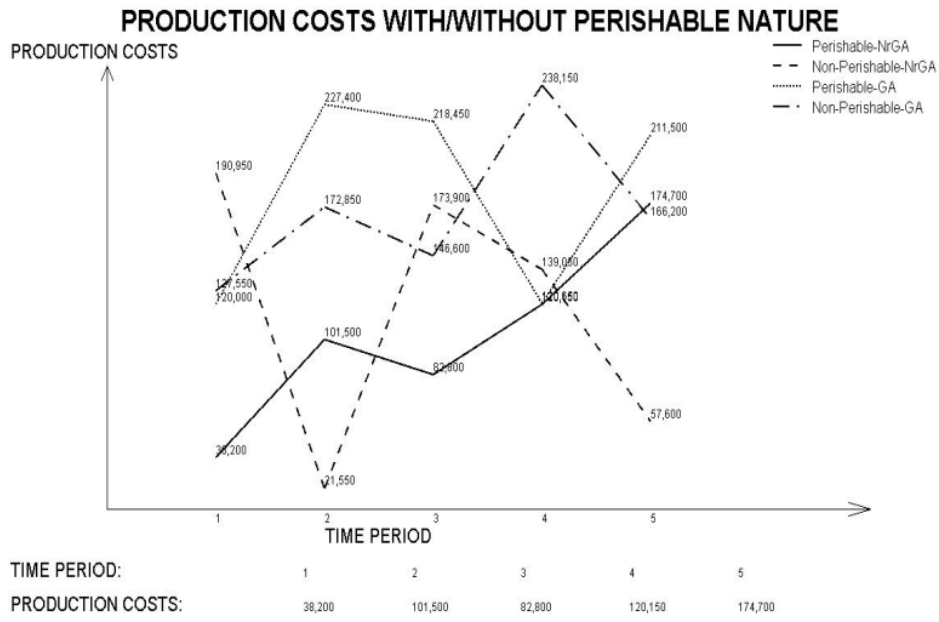


Figure 12. Production, Ordering and Handling costs

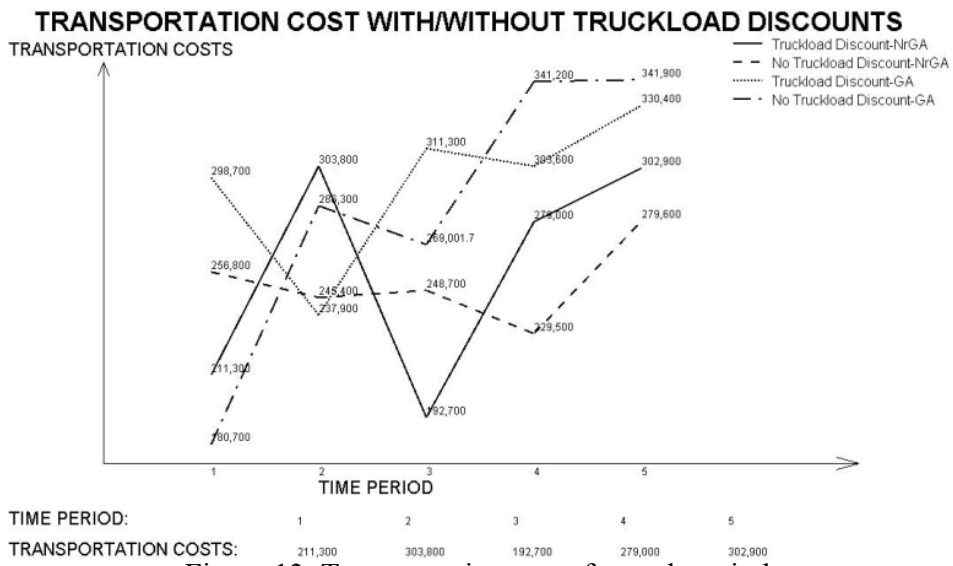


Figure 13. Transportation costs for each period

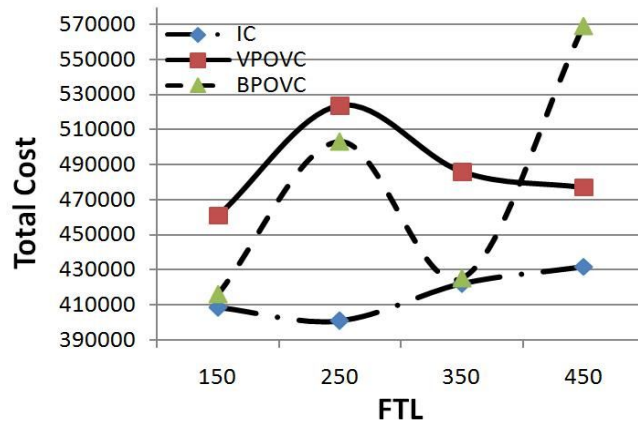


Figure 14. Average Total cost per period against FTL in centralized and decentralized chain

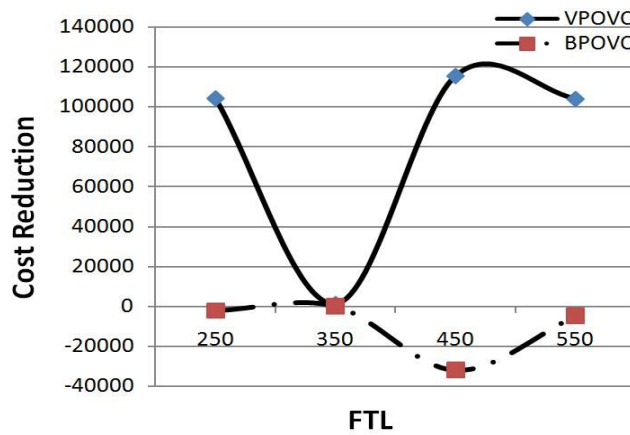


Figure 15. Average Cost reduction per period against FTL for a move from BPOVC to IC

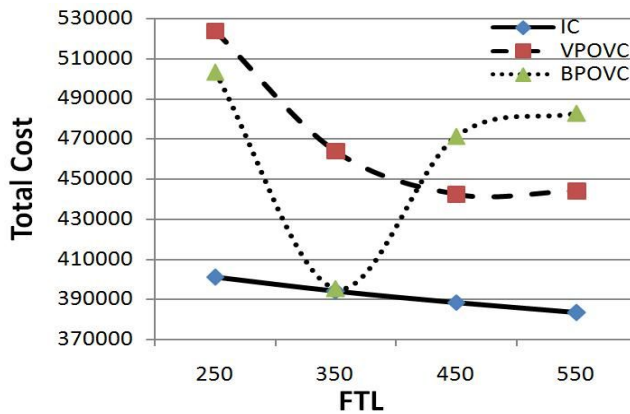


Figure 16. Average Total cost per period against FTL in centralized and decentralized chains (Fleet selection)

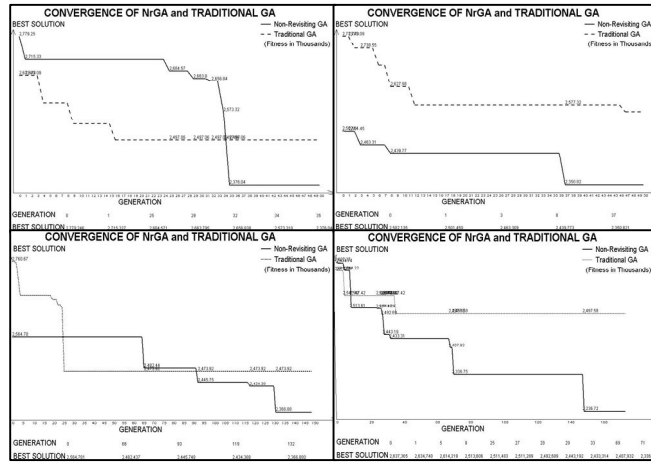


Figure 17. Convergence of NrGA and GA for population-generation combinations: 50-30, 50-50, 50-150 and 65-175

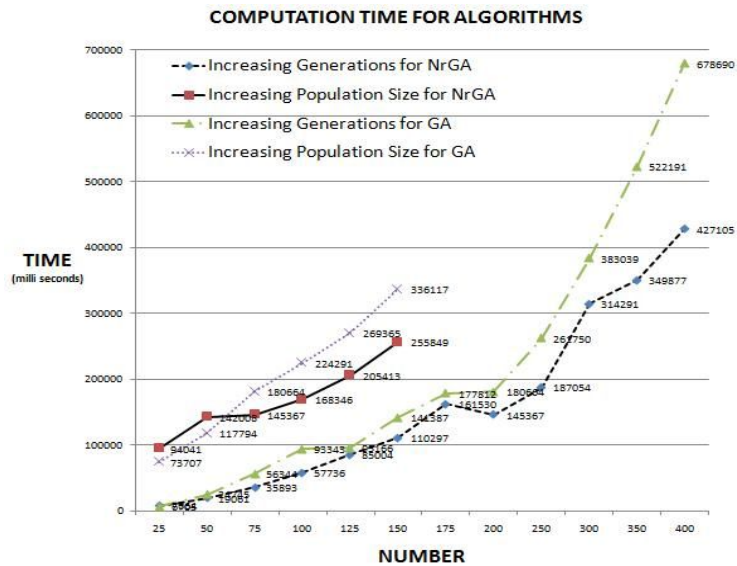


Figure 18. Average Computation Time for NrGA and GA

TABLE I
DATA FOR THE PROBLEM

(a) Demand

Market \ Period	1	2	3	4	5
1	55	65	45	55	40
2	50	58	75	85	30
3	65	70	55	70	45
4	75	85	50	65	70

(b) Warehouse-Market Distance Matrix

Warehouse \ Market	1	2	3	4
1	18	12	10	10
2	10	9	9	14

(c) Plant-Warehouse Distance Matrix

Plant \ Warehouse	1	2
1	10	12

FTL = 250 units; HTL = 125 units; A = 0.025 unit fraction per unit time; B = 0.720 periods.

TABLE II

ALGORITHM PARAMETERS	
Crossover Rate	0.09
Mutation Rate (for GA)	0.01
Population size	50
Maximum Generations	175
Chromosome Length	475
Gene Length	1

TABLE III

COMPUTATIONAL RESULTS FOR A SIMULATION RUN OF 10 TRIALS

EXPER- IMENT	PRODUCT-ION, HANDLING & ORDERING COSTS	INVENTORY HOLDING AND LOST COSTS	TRANSPORT- ATION COSTS	TOTAL FITNESS COST	VALUE	COMPUTA- TION TIME (milli-sec)
NrGA						
1.1	576600	1447	1051700	1629747	2458631	21072
1.2	686950	1227	1092500	1780677	2636570	23162
1.3	681250	2879	1179100	1863229	2655782	24153
1.4	668450	2110	1043900	1714460	2648309	20794
1.5	618200	1444	1248300	1867944	2651190	21868
1.6	632500	3530	948800	1584830	2495136	22654
1.7	640200	743	1068700	1709643	2504634	23252
1.8	501550	97	1254800	1756447	2560202	21975
1.9	625850	726	1107300	1733876	2528232	22467
1.10	586450	1714	1169200	1757364	2566171	21514
Traditional GA						
1.1	547300	179	1229800	1777279	2569683	139102
1.2	639450	1377	1074500	1715327	2596386	130909
1.3	497950	763	1172943	1671656	2612990	90838
1.4	529350	2168	1219763	1751282	2501085	91530
1.5	536500	3083	1275500	1815083	2667401	91511
1.6	413550	239	1390300	1804089	2605871	91597
1.7	477950	1524	1320693	1800167	2644977	91187
1.8	702404	2874	1027648	1732927	2620069	93072
1.9	617950	4161	1211900	1834011	2614466	91109
1.10	558724	3323	1222956	1785004	2580722	94510

TABLE IV

$\Gamma/[Tc]$ AND Δ FOR DIFFERENT S_{KT}
 (S_{KT} IS ASSUMED TO BE SAME FOR ALL MARKETS AND ALL PERIODS)

S_{KT}									
8		9		10		11		12	
$\Gamma/[Tc]$	Δ	$\Gamma/[Tc]$	Δ	$\Gamma/[Tc]$	Δ	$\Gamma/[Tc]$	Δ	$\Gamma/[Tc]$	Δ
1.00	0.02	1.01	0.03	0.99	0.04	1.00	0.05	1.00	0.03
0.98	0.02	1.01	0.02	0.98	0.05	1.01	0.01	0.97	0.06
1.01	0.02	0.94	0.05	1.01	0.04	1.01	0.07	1.01	0.04
1.05	0.15	0.99	0.05	1.00	0.05	1.00	0.02	1.00	0.03
1.01	0.03	1.01	0.01	0.98	0.04	0.95	0.01	0.99	0.03

TABLE V

VARIATION IN AVERAGE TOTAL COST (LOW DEMAND)

FTL	f_f	f_h	[Tc]
150	450	300	508234
250	600	400	501071
350	750	500	522920
450	900	600	532791

TABLE VI

VARIATION IN AVERAGE TOTAL COST (HIGH DEMAND)

FTL	F_f	F_h	[Tc]
350	750	500	623643
450	900	600	593410
550	1050	700	609285
650	1200	800	627691

TABLE VII

VARIATION IN TOTAL COST PER PERIOD WITH LEAD TIME

lt	[Tc]
0	300000
1	373526
2	399518
3	429625

TABLE VIII

COMPARISON OF RESULTS WITH CPLEX

Problem Size (Plants x Cross- docks x Markets x Periods)	Total Supply Chain Cost		
	NrGA	GA	CPLEX MIP
1x1x1x2	46,587	116,800	35,661
1x1x1x3	116,212	142,608	64,084
1x1x1x4	287,787	281,937	87,212
1x1x1x5	318,712	403,650	99,466
1x1x2x2	117,237	191,937	65,454
1x1x2x3	212,412	224,699	73,415
1x2x2x2	127,212	144,975	76,246
1x2x2x3	242,244	421,584	87,952