SciencePG
Science Publishing Group

A Hybrid Genetic Algorithm-Simulated Annealing for Integrated Production-Distribution Scheduling in Supply Chain Management

Setareh Abedinzadeh[1], Hamid Reza Erfanian[2, *], Mojtaba Arabmomeni[3]

[1]Department of Industrial Engineering, University of Science and Culture, Tehran, Iran

[2]Department of Mathematics, University of Science and Culture, Tehran, Iran

[3]Department of Industrial Engineering, University of Science and Technology, Tehran, Iran

Email address:

erfanian@usc.ac.ir (H. R. Erfanian)

[*]Corresponding author

To cite this article:

Setareh Abedinzadeh, Hamid Reza Erfanian, Mojtaba Arabmomeni. A Hybrid Genetic Algorithm-Simulated Annealing for Integrated Production-Distribution Scheduling in Supply Chain Management. *International Journal of Theoretical and Applied Mathematics*. Vol. 3, No. 6, 2017, pp. 229-238. doi: 10.11648/j.ijtam.20170306.19

Abstract: In this paper, we present an integrated production-distribution (P-D) model which considers rail transportation to move deteriorating items. The problem is formulated as a mixed integer programming (MIP) model, which could then be solved using GAMS optimization software. A hybrid genetic algorithm-simulated annealing (GA-SA) is developed to solve the real-size problems in a reasonable time period. The solutions obtained by GAMS are compared with those obtained from the hybrid GA-SA and the results show that the hybrid GA-SA is efficient in terms of computational time and the quality of the solution obtained.

Keywords: Integrated Production-Distribution Planning, Rail Transportation, Deteriorating Items, Scheduling, Hybrid Genetic Algorithm-Simulated Annealing

1. Introduction

A supply chain (SC) may be considered as an integrated process in which a group of several organizations, such as suppliers, producers, distributors and retailers, work together to acquire raw materials with a view to converting them into end products which they distribute to retailers [1]. The two core optimization problems in a SC are production and distribution planning. In the production planning, decisions such as hiring and firing of labors, regular time and overtime production, subcontracting and machine capacity levels are made for a definite planning horizon (i.e. usually a one year period). Whereas, in the distribution planning decisions relates to determining which facility (ies) would supply demands of which market(s) [2].

Unlike traditional supply chains with members competing to reduce their individual costs, the overall cost of the entire supply chain is minimized in a cooperative supply chain. The savings from cooperation may be shared among the members, while a lower average cost and a lower cost variation is materialized for individual members.

In this study, a multiple-product, multiple time-period P-D model with consideration of rail transportation to move deteriorating items is developed. Thus, we briefly review the related literature of the mentioned problem. The followings are the previous studies on multiple-product, multiple time-period. Saracoglu et al, formulated an approach for multi-product multi-period (Q, r) inventory models that calculates the optimal order quantity and optimal reorder point under the constraints of shelf life, budget, storage capacity, and "extra number of products" promotions according to the ordered quantity [15]. First, the problem is modelled as an integer linear programming (ILP) model. Next, a genetic algorithm (GA) solution approach with an embedded local search is proposed to solve larger scale problems. The results indicate that the

proposed approach yields high-quality solutions within reasonable computation times. Pasandideh et al, considered a bi-objective optimization of a multi-product multi-period three-echelon supply-chain-network problem [13]. The network consists of manufacturing plants, distribution centers (DCs), and customer nodes. The majority of the parameters in this network including fixed and variable costs, customer demand, available production time, set-up and production times, all are considered stochastic. The goal is to determine the quantities of the products produced by the manufacturing plants in different periods, the number and locations of the warehouses, the quantities of products transported between the supply chain entities, the inventory of products in warehouses and plants, and the shortage of products in periods such that both the expected and the variance of the total cost are minimized. The problem is first formulated into the framework of a single-objective stochastic mixed integer linear programming model. Then, it is reformulated into a bi-objective deterministic mixed integer nonlinear programming model. To solve the complicated problem, a non-dominated sorting genetic algorithm (NSGA-II) is utilized next. As there is no benchmark available in the literature, another GA-based algorithm called non-dominated ranking genetic algorithm (NRGA) is used to validate the results obtained. Some numerical illustrations are provided to show the applicability of the proposed methodology and select the best method using a t-test along with the simple additive weighting (SAW) method. Priyan et al, considered a distributor and a warehouse consisting of a serviceable part and a recoverable part supply chain problem in which there are several products, the distributor has limited space capacity and budget to purchase all products [14]. A mathematical model is employed to optimize the order quantity, lead time and total number of deliveries with the objective of minimizing system total cost. A simple Lagrangian multiplier technique to solve the model which is a constrained non-linear programming. At the end, numerical and sensitivity analysis are given to show the applicability of the proposed model in real-world product returns inventory problems.

In addition, rail transportation has been widely used in supply chain management (SCM). Yaghini et al, presented a comprehensive review of multi-commodity network design (MCND) problems modeling, their applications in rail freight transportation planning, and solution methods which have been developed to solve them [16]. Hajiaghaei-Keshteli et al, proposed an integrated production and transportation model, which considers rail transportation to deliver orders from a facility to the customers [5]. The problem is to determine both production schedule and rail transportation allocation of orders to optimize customer service at minimum total cost. Different destinations of the trains, trains' capacities, and different transportation costs are the main aspects of the work which are considered. A heuristic, two meta-heuristics and some related procedures are developed to cope with the NP-hardness of the problem. Besides, Taguchi experimental

design method is utilized to set and estimate the proper values of the algorithms' parameters to improve their performance. For the purpose of performance evaluation of the proposed algorithms, various problem sizes are employed and the computational results of the algorithms are compared with each other. Besides, Hajiaghaei-Keshteli et al, developed the integrated scheduling of production and rail transportation. The problem is to determine both production schedule and rail transportation allocation of orders to optimize customer service at minimum total cost [6]. Some procedures and heuristics are utilized to encode the model in order to address it by two capable meta-heuristics: GA, and Keshtel algorithm (KA). Taguchi experimental design method is utilized to set and estimate the proper values of the algorithms' parameters to improve their performance. Besides, various problem sizes are employed and the computational results of the algorithms are compared with each other.

Over the past few days, there have been fruitful efforts for considering deteriorating items in SCM. Ghiami et al, regarded a single-manufacturer, multi-buyer model for a deteriorating item with finite production rate [3]. The authors relaxed the assumption on the production capacity and found the average inventory of the supplier. It is shown that in case the production rate is not high, the existing models may not be sufficiently accurate. Also a sensitivity analysis is conducted to show how the model reacts to changes in parameters. Maihami et al, considered the problem of replenishment policy and pricing for non-instantaneous deteriorating items subject to promotional effort [12]. A price dependent stochastic demand function in which shortages are allowed and partially backlogged, is adopted. The major objective is to simultaneously determine the optimal selling price, the optimal replenishment schedule, and the optimal order quantity to maximize the total profit. An algorithm is employed obtain the optimal solution. Finally, the numerical example is extended by performing a sensitivity analysis of the model parameters and discusses specific managerial insights. Zhang et al, developed a one-manufacturer-one-retailer supply chain model for deteriorating items with controllable deterioration rate and price-dependent demand in which both players cooperatively invest in preservation technology to reduce deterioration [17]. Algorithms are designed to obtain the pricing and preservation technology investment strategies in both integrated and decentralized scenarios. Numerical simulations and sensitivity analysis of the equilibrium strategies and coordinating results on key system parameters are given to verify the effectiveness of the contract, and meanwhile get some managerial insights.

This paper is organized as follows. Section 2 explains the characteristics of the problem; in Section 3, the mathematical model is developed and Section 4 introduces a hybrid GA-SA to solve the model; computational results and discussion are included in Section 5. Finally, conclusions for future research are presented in Section 6.

2. The Problem Definition and Assumptions

In this section, we explain the definition of the problem and basic assumptions to better understand the problem. As mentioned before, in this paper we consider a multiple-product, multiple time-period production-distribution model in SCM. The aim is to make optimal decisions about the optimal amount of products in each time-period, optimal sequence of job processing in production line to prepare final products, the way of using available facilities to transport products from production line to customers with consideration of constraints (i.e. warehouse capacity, transportation, customer satisfaction, perish ability of products). Strategic decisions (i.e. routing, determining the railway) and operational decisions (i.e. amount of production and storage) are also regarded.

The considered assumptions of the problem are as follows:
- Transportation of products is performed using rail fleet.
- Developing the rail transportation network using the construction of new rails during the period of decision making is not allowed.
- It is possible to change rail roads in each period.
- The number and the capacity of the trains are predetermined.
- The scheduling system considered in the production line, is single-machine.
- Products can be stored in the warehouses of the production part.
- Products are deteriorating items which have different life time.
- Delivery time of products to customers are limited and predetermined.
- Penalty cost is considered when products deliver to customer after the determined due date.

3. The Mathematical Model

The following notations are used for mathematical formulation of the proposed model:

Indices:

i,j	The index of job/product (in fact, each product is related to the specific job, so the index of job and product are the same)	$(1 \leq i \leq N)$
l	The index of rail road	$(1 \leq l \leq L)$
k	The index of train	$(1 \leq k \leq K)$
$c, \hat{c}$	The index of customer	$(1 \leq c, \hat{c} \leq C)$
$t, \hat{t}$	The index of time period	$(1 \leq t, \hat{t} \leq T)$

Parameters:

$dis_{\hat{c}c}^{train}$	Rail distance between customer c and customer $\hat{c}$
dem_{jct}	Demand of customer c for product j in period t
cap^{train}	Capacity of train transportation
cap^{inv}	Capacity of storing products
tm_j^{proc}	Enough time for processing a unit of product j
vol_j	Volume of product j
dd_{jct}	Delivery time of product j to customer c in period t
ed_j	The time period before expiry date of product j
c_{jct}	Penalty for delivery of product j to customer c in period t
η	The cost of train transportation in the distance
M	A big number

tm_{cc}^{travel} $\left(\sum_c dis_{1c}^{train} = \sum_c dis_{c1}^{train} = 0 \right)$ Time period between customer c and customer $\hat{c}$

Variables:

α_{cc}	Binary variable which is equal to 1 if there is a rail road that connects customer c to customer $\hat{c}$
λ_{klt}	Binary variable which is equal to 1 if train k is assigned to rail road l in period t
$z_{c\hat{c}lt}$	Binary variable which is equal to 1 if rail road l leaves customer c to visit customer $\hat{c}$ in time period of t
x_{ijt}	Binary variable which is equal to 1 if job i is processed before job j in period t
β_{ikt}	Binary variable which is equal to 1 if train k transport product i in period t
γ_{clkt}	Binary variable which is equal to 1 if train k which is belonged to rail road l, passes customer c in period t
μ_{cclt}	Number of assigned trains to rail road l which leave customer c to visit customer $\hat{c}$ in period t

st_{jt}	Initialization time of processing the job j in period t
ct_{jt}	Termination time of processing the job j in period t
N_{jt}	The number of product j which is processed in period t
y_{jckt}	The number of product j which are delivered by train k to customer c in period t
inv_{jt}	Inventory of product j in the warehouse in period t
dt_{ckt}	The time that train k leaves customer c in period t
dlt_{jct}	Amount of tardiness in delivery of product j to customer c in period t
LC_{jct}	Penalty for delivery of product j to customer c in period t

The mathematical formulation of the proposed model is as follows:

$$Min\ cost = \sum_{j,c,t} LC_{jct} + \eta \times \sum_{\hat{c},c,l,t} (dis_{\hat{c}c} \times \mu_{\hat{c}clt})$$

s.t:

$$\sum_j x_{ijt} = \sum_j x_{jit} \quad \forall i,t \tag{1}$$

$$\sum_j x_{ijt} = 1 \quad \forall i,t \tag{2}$$

$$ct_{jt} = st_{jt} + N_{jt} \times tm_j^{proc} \quad \forall j,t \tag{3}$$

$$st_{jt} + M(1 - x_{ijt}) \geq ct_{it} \quad \forall i, j > 1, t \tag{4}$$

$$\sum_{\hat{c}} z_{\hat{c}clt} = \sum_{\hat{c}} z_{c\hat{c}lt} \quad \forall c,l,t \tag{5}$$

$$\sum_{\hat{c},l} z_{\hat{c}clt} = 1 \quad \forall c,t \tag{6}$$

$$dt_{1kt} + M(1 - \beta_{jkt}) \geq ct_{jt} \quad \forall j,k,t \tag{7}$$

$$dt_{ckt} + M(2 - z_{\hat{c}clt} - \lambda_{klt}) \geq dt_{\hat{c}kt} + tm_{\hat{c}c}^{travel} \quad \forall \hat{c}, c > 1, k, t \tag{8}$$

$$inv_{j1} = N_{j1} - \sum_{c,k} y_{jck1} \quad \forall j \tag{9}$$

$$inv_{jt} = inv_{j(t-1)} + N_{jt} - \sum_{c,k} y_{jckt} \quad \forall j, t > 1 \tag{10}$$

$$inv_{jt} \leq \sum_{c,k,\hat{t}=t+1}^{t+ed_j} y_{jck\hat{t}} \quad \forall j, t \leq T - ed_j \tag{11}$$

$$inv_{jt} \leq \sum_{c,k,\hat{t}=t+1}^{T} y_{jck\hat{t}} \quad \forall j, t > T - ed_j \tag{12}$$

$$dlt_{jct} + M(1 - \beta_{jkt}) \geq dt_{ckt} \quad \forall j,c,k,t \tag{13}$$

$$LC_{jct} \geq (dlt_{jct} - dd_{jct}) \times c_{jct} \quad \forall j,c,t \tag{14}$$

$$y_{jckt} \leq M \times \beta_{jkt} \quad \forall j,c,k,t \tag{15}$$

$$\sum_j y_{jckt} \leq M \times \sum_l \gamma_{clkt} \quad \forall c,k,t \tag{16}$$

$$\gamma_{clkt} \leq \frac{\lambda_{klt} + \sum_{\hat{c}} z_{\hat{c}clt}}{2} \quad \forall c,l,k,t \tag{17}$$

$$\sum_l \lambda_{klt} \leq 1 \quad \forall k,t \tag{18}$$

$$\sum_{l,t} z_{\hat{c}clt} \leq M \times \alpha_{\hat{c}c} \quad \forall c,\hat{c} \tag{19}$$

$$\sum_k y_{jckt} \geq dem_{jct} \quad \forall j,c,t \tag{20}$$

$$\sum_j inv_{jt} \times vol_j \leq cap^{inv} \quad \forall t \tag{21}$$

$$\sum_{j,c} y_{jckt} \times vol_j \leq cap^{train} \quad \forall k,t \tag{22}$$

$$\mu_{\hat{c}clt} + M(1 - z_{\hat{c}clt}) \geq \sum_k \lambda_{klt} \quad \forall \hat{c},c,l,t \tag{23}$$

$$st_{jt}, ct_{jt}, dt_{ckt}, dlt_{jct}, LC_{jct} \geq o \quad \forall j,t,c,k \tag{24}$$

$$\alpha_{\hat{c}c}, \lambda_{klt}, z_{\hat{c}clt}, x_{ijt}, \beta_{ikt}, \gamma_{clkt} = \{0,1\} \quad \forall c,\hat{c},i,t,j,k,l. \tag{25}$$

The objective function minimizes the transportation cost, as well as tardiness cost of product delivery. The processing condition of each job and the sequence of processing jobs are represented in constraint (1) and constraint (2), respectively. Constraint (3) and constraint (4) expresses the initialization time and termination time of jobs, respectively. Constraint (5) guarantees that if a rail road is entered to a place, it must be leaved there. Constraint (6) assumes that each customer must be visited only by a rail road. The time which trains leave the production part and customers are shown in constraint (7) and constraint (8), respectively. Constraint (9) and constraint (10) delineate the relationship between inventories. Constraint (11) and constraint (12) show that products are deteriorating items. Amount of tardiness from due date in delivery of products and penalty are depicted in constraint (13) and constraint (14), respectively. According to the constraints (15-17), it is possible to deliver a product to customer c by train k if the train k transport the mentioned product from its production part and the rail road which is related to the mentioned train visits customer c. constraint (18) explains that it is not permitted to assign a train to more than one rail road during a period of time. Constraint (19) expresses that it is not possible to visit the rail road which

connects node (customer) c to node (customer) $\hat{c}$, unless the rail road between these two nodes is made. Constraint (20) assures the satisfaction of demand. Constraint (21) and constraint (22) explains that the capacity of storage and the capacity of transportation must not be exceeded from the considered limit. The numbers of allocated trains to rail road l in time period t which leave customer c to customer $\hat{c}$ are represented in constraint (23). Constraints (24-25) represent the range of model variables.

4. Hybrid GA-SA Algorithm

4.1. Simulated Annealing

SA which was firstly developed by Kirkpatrick et al, is considered as a local search algorithm [9]. It is based on the analogy between the process of finding a possible best solution of a combinatorial optimization problem and the annealing process of a solid to its minimum energy state in statistical physics. SA is similar to hill climbing or gradient search with a few modifications but it does not require the optimization function to be smooth and continuous. The SA algorithm is an iterative search procedure based on a neighborhood structure. The quality of the annealing solution is sensitive to the way of selecting the candidate (trial) solutions. Thus, a neighborhood structure, including a generation mechanism and its set size, is crucial for the performance of the SA algorithm. The SA algorithm with a larger neighborhood performs better than that with a smaller one, i.e., a larger neighborhood makes it likely to reach out over a much broader space of the solution set. The neighborhood structure provides global asymptotic convergence for an arbitrary solution. Hence, there exists at least one global optimal solution that can be reached in a finite number of iterative transitions. The process of searching begins with one initial random solution. A neighborhood of this solution is generated using a neighborhood move rule and then the cost between neighborhood solution and current solution can be found, according to Eq. (26).

$$\Delta C = C_i - C_{i-1} \qquad (26)$$

Where ΔC , C_i and C_{i-1} represent change amount between costs of the two solutions, neighborhood solution and current solution, respectively. If the cost declines, the current solution is replaced by the generated neighborhood solution. Otherwise, the current solution is replaced with the new neighborhood solution with some probability, which is generated using Eq. (27) and the same steps are repeated. After the new solution is accepted, inner loop is checked. If the inner loop criterion is met, the value of temperature is decreased using a predefined cooling schedule. Otherwise, a new neighborhood solution is regenerated and the same steps are repeated. The process of searching is repeated until the termination criteria are met or no further improvement can be found in the neighborhood of the current solution. The termination criterion (outer loop) is predetermined:

$$e^{(-\Delta C/T)} \succ R \qquad (27)$$

where the temperature (T) is a positive control parameter. R is a uniform random number between 0 and 1. SA operators are described as follows:

4.1.1. Neighborhood Move

First, an initial solution randomly is generated. Then, neighborhood move is used to produces solution close to the current solution in search space. Basically, two neighborhood moves are employed: swapping move and shifting move [10]. In swapping move, two genes are randomly selected and the positions of these genes were swapped. Then, a new neighborhood solution is produced. In shifting move, two genes are randomly selected similarly and the second gene is put in front of other genes. Thus, a new solution is produced.

4.1.2. Cooling Schedule

Each problem requires a unique cooling schedule and it becomes very difficult to pick the most appropriate schedule within a few simulations. These schedules are proportional decrement schedule and Lundy and Mees schedule [11]. In proportional decrement schedule, the relationship between the temperature values in kth and $(k+1)$th iterations is according to Eq.(28)

$$T_{k+1} = \alpha T_k \qquad \alpha = \sqrt[M]{\frac{T_f}{T_i}} \qquad (28)$$

where T_k and T_{k+1} are temperatures in kth and $(k+1)$th iterations, respectively and α is the coefficient between two temperatures that varies between 0 and 1. Besides, M, T_f and T_i are the number of iteration, the final and the initial temperatures, respectively. In Lundy and Mees schedule, the relationship between T_{k+1} and T_k is according to Eq. (29)

$$T_{k+1} = \frac{T_k}{1 + \beta T_k} \qquad \beta = \frac{T_i - T_f}{MT_iT_f} \qquad (29)$$

$\beta \succ 0$ is the coefficient between two temperatures: T_{k+1} and T_k .

4.1.3. Inner Loop and Outer Loop Criterion

Inner loop criterion determines the number of possible new solutions to produce in each temperature value and outer loop criterion is used to stop the searching process.

4.2. Genetic Algorithms

GA, one of the optimization and global search methods, is based on the simulated natural selection [4]. GA was developed by Holland in the 1970s. It is applied effectively to solve various combinatorial optimization problems and is based on probabilistic rules [7]. Selection, crossover and mutation are the most essential genetic operators. GA searches

new and better solutions to a problem by improving current population. The search is guided towards the principle of the survival of the fittest. This is obtained by extracting the most desirable characteristics from a generation and combining them to form the next generation. The population includes a set of chromosomes. Each chromosome in the population is a possible solution. The quality of each possible solution is measured by fitness function. First, GA generates initial population and then calculates the fitness value according to fitness function for each chromosome of the population. Fitness function is specifically generated for each problem. It may be simple or complex according to the problem. Then optimization criterion is checked. If the optimization criterion is met, this solution can be considered as the best solution. Otherwise, new population is regenerated using GA operators (selection, crossover, and mutation). According to their fitness values, chromosomes are selected for crossover operation using a selection operator. Therefore each chromosome will contribute to the next generation in proportion to its fitness. Then crossover and mutation operators are applied to the selected population to create the next population. The process continues through generations until the convergence on optimal or near-optimal solutions. However GA cannot guarantee to find the best optimal solution. GA operators are described as follows:

4.2.1. Population

It is a set of possible solutions to the problem. Since the size of the population varies according to problem, there is no clear mark on how large it should be. Then, fitness value for each chromosome of the population is calculated according to fitness function.

4.2.2. Elitist Selection

Selection operator selects the chromosomes to be mated according to their fitness values. Elitist selection is used here which means that a practical variant of the general process of constructing a new population is to allow the best organism(s) from the current generation to carry over to the next, unaltered. This guarantees that the solution quality obtained by the GA will not decrease from one generation to the next.

4.2.3. Crossover

Crossover operator is a powerful tool for exchanging information between chromosomes and creating new solutions. It is expected that good parents may produce better solutions.

4.2.4. Mutation

This operator is used to prevent reproduction of similar type chromosomes in population. Mutation operator randomly selects two genes in chromosome and swaps the positions of these genes to produce a new chromosome. This technique is called swap mutation.

4.3. Hybrid Algorithm

Hybridization refers to combining two search algorithms

to solve a given problem. This is often a population-based search such as GA with local searches performed by other algorithms like simulated annealing, greedy algorithm, etc. The main drawbacks of SA algorithms are the computation time and limited convergence behavior. For better results the cooling has to be carried out very slowly and this significantly increases the computation time. Various computations in SA operators increase the computation time when the dimension of the problem grows. Optimum iteration should be selected to decrease computation time. With this iteration, hybrid approach is needed to obtain the global optimum solution. It is common to start SA from a random configuration. The performance of SA may be improved if more information is known about the problem in hand.

Hence it might be better to start from a configuration which is a good local minima, like a configuration obtained by a GA algorithm search. Starting from a good local minimal with initial high temperature will provide an opportunity to escape the poor local minima and attain a better solution, possibly global minima [7, 8]. This paper uses a hybrid scheme to integrate P-D scheduling in SCM using GA and SA. GA can reach the region near an optimum point relatively quickly, but it can take many function evaluations to achieve convergence. A technique used here is to run GA for a small number of generations to get near an optimum point. Then the solution from GA is used as an initial point for SA that is faster and more efficient for local search.

5. Computational Results

This section explains the test problems which aim at showing the applicability of the proposed algorithm. The computational results are reported, evaluated and analyzed with respect to the proposed model. For the small and medium sized problems, the solutions presented by hybrid GA-S are compared with the results obtained from GAMS optimization software.

5.1. Designing the Test Problems

Various test problems, with different sizes are considered to assess the performance of the proposed algorithm. We consider three sets of 8 small sized, 8 medium sized and 3 large sized problems to be solved using hybrid GA-S, i.e., a total of 19 instances were run.

In each problem, the values of each group of parameters are generated randomly between their lower and upper bounds, based on Table 1.

5.2. Setting the Hybrid GA-SA Parameters

Parameters of the hybrid GA-SA include *population size*, *cross over rate*, *mutation rate*, and *T*. Primary tests are carried out in order to determine the values of these parameters. A trade-off between the solution time and the quality determined the appropriate value. The values for

population size, *cross over rate*, *mutation rate*, and *T* are set to 200, 0.3, 0.15 and 100 respectively.

5.3. Computational Results

The proposed P-D model has been solved using CPLEX solver of GAMS for small and medium sized problem. For large sized problem, the hybrid GA-SA is employed. The proposed algorithm is coded in Matlab. All the test problems are solved on an Intel corei5 computer with 4 GB RAM and 2.67 GHz CPU.

A quality criterion, ERROR, is defined to show the deviation of the value of the DE solutions from the values of GAMS, according to Eq. (30).

$$ERROR = \frac{(GASA.Z - GAMS.Z)}{GAMS.Z} \qquad (30)$$

To investigate the performance of GAMS optimization software and hybrid GA-SA algorithm, two criterions, i.e., objective value and run time have been considered. The results of small and medium sized problem are shown in Table 2 and Table 3, respectively.

Based on the results of Table 2, small sized problem are applicable to the problem in which the run time of GAMS optimization software is less than 50s. On average, hybrid GA-SA achieved 96% of the exact optimal solutions within 17% of the exact run time. In other word, the average deviation from the optimum for small sized problem does not exceed 4% of error. The trivial deviation shows the efficiency of the proposed algorithm.

Figure 1 depicts hybrid GA-SA runtime versus GAMS run time for small sized problem. It is shown that GAMS run time increases exponentially as the size increases.

Figure 2 delineates hybrid GA-SA objective value versus GAMS objective value for small sized problem. It is clear that hybrid GA-SA objective value compared to that of GAMS decreases when the problem size increases.

Base on the results of Table 3, in medium sized problem the run time of GAMS optimization software is less than 150s. As it is obvious, hybrid GA-SA algorithm achieved 89% to 92% of the exact optimal solution, within 9% to 18% exact run time. This result denotes the efficiency of the proposed algorithm.

Figure 3 shows, hybrid GA-SA run time versus GAMS run time for medium sized problem. It is observed that the run time and complexity of solving problems by hybrid GA-SA are lower than GAMS optimization software. Besides, the differences between the run time of GAMS optimization software and hybrid GA-SA is increased, as the problem size accelerates. This means that hybrid GA-SA solves the large sized problem in an acceptable time.

Figure 4 delineates hybrid GA-SA objective values versus GAMS objective value for medium sized problem. The trivial deviation shows the high accuracy of the proposed algorithm to cope with large sized problems.

Objective value and run time for GAMS optimization software and hybrid GA-SA for large sized problem are calculated. As shown in Table 4, out of memory message of GAMS appears regarding test problem 19. This is due to the saturation of RAM and stopping the calculation before achieving an acceptable solution.

We also considered 1000s as a limit to stop the algorithm after a specific run time. As it is clear, an error message is appeared in test problem 17 and 18 run by GAMS, which means that the run time exceeded 1000s and GAMS is not able to solve the problem. Furthermore, the proposed hybrid algorithm is not able to converge in considered limit but this algorithm achieved acceptable solution in all 3 test problems. It is observed that, the deviation of hybrid GA-SA from the optimum for test problems 17 and 18 are 8% and 12%, respectively. This denotes the high capability of meta-heuristic algorithms for solving the large sized problem.
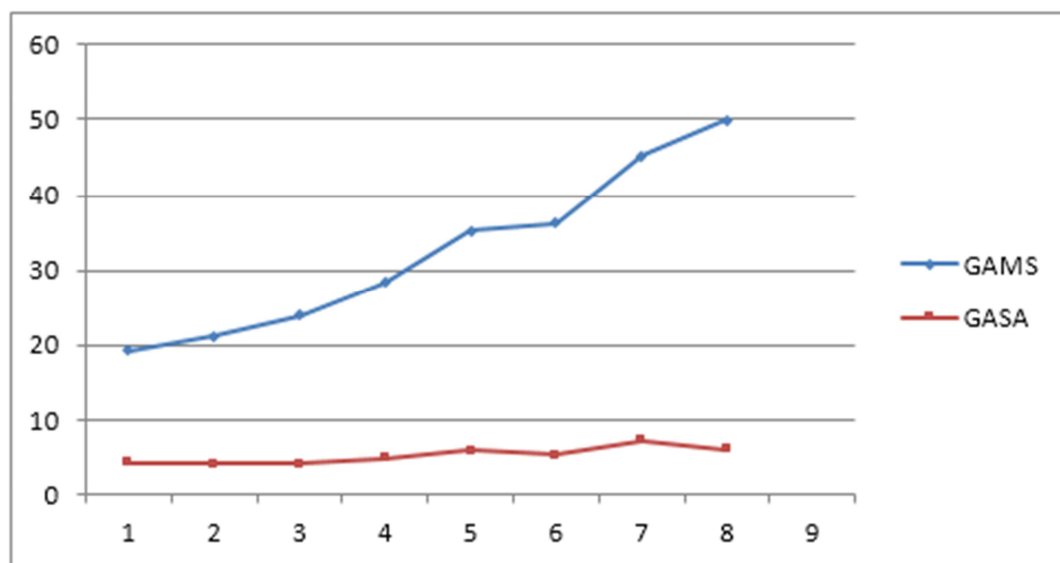

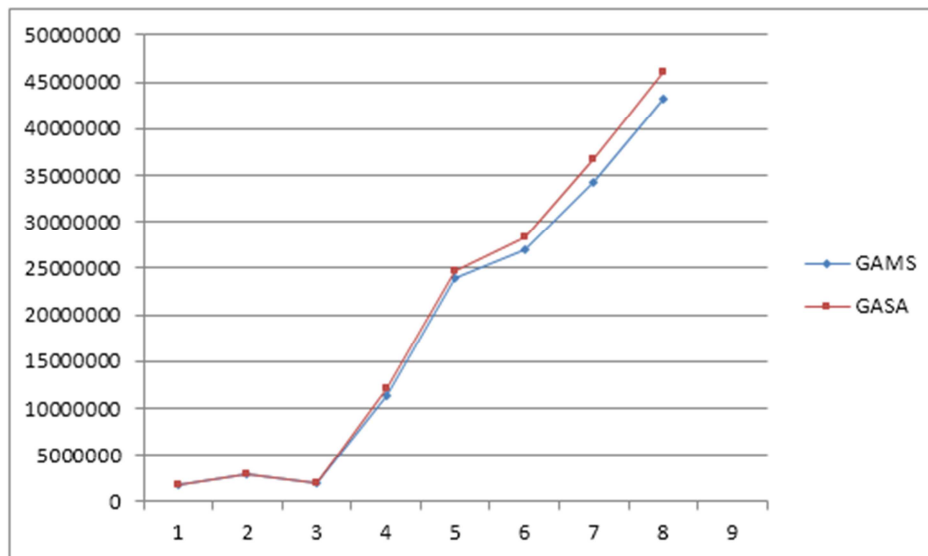*Figure 1. Hybrid GA-SA run time versus GAMS run time for small sized problem.*

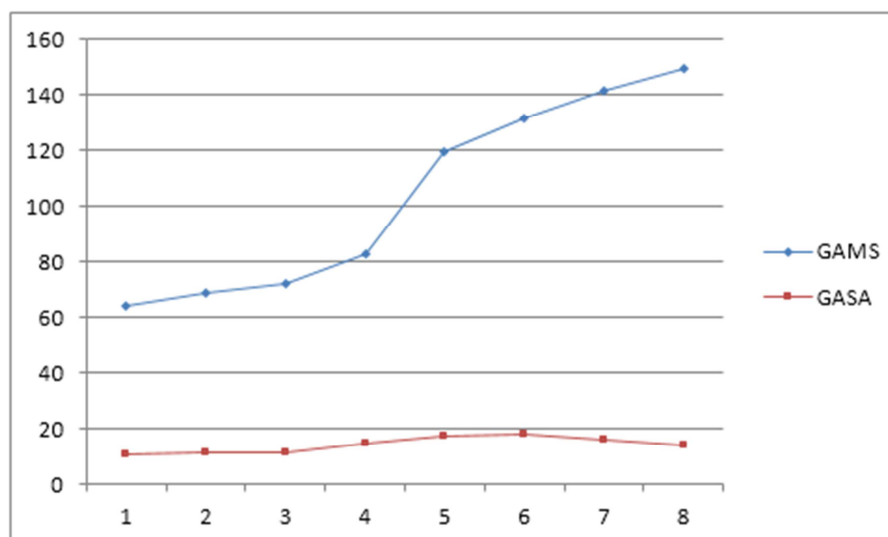**Figure 2.** *Hybrid GA-SA objective value versus GAMS objective value for small sized problem.*

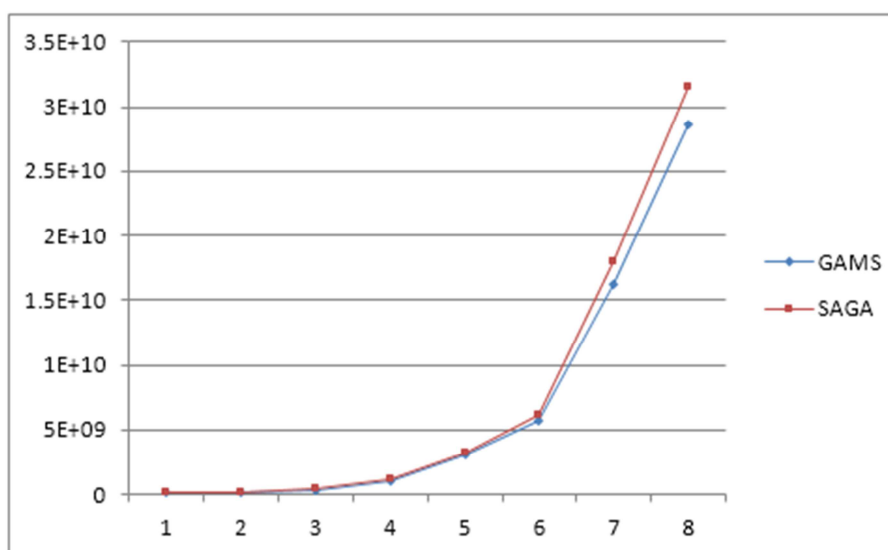**Figure 3.** *Hybrid GA-SA run time versus GAMS run time for medium problem.*

Figure 4. *Hybrid GA-SA objective value versus GAMS objective value for medium sized problem.*

Table 1. *Some of important inputs for the model.*

Inputs	Value
$\alpha_{c\hat{c}}$	Bernoulli $(0.7,\{c,c\})$
Node position of x	continuous uniform$([0\ 200],\{c\})$
Node position of y	continuous uniform$([0\ 200],\{c\})$
$dis_{\hat{c}c}^{train}$	Euclidean distance
$tm_{\hat{c}c}^{travel}$	$dis_{ij}\times$continuous uniform$([1\ 3],\{c,c\})$
dem_{jct}	Ceil(continuous uniform$([1\ 5],\{j\})\times$ continuous uniform$([10\ 200],\{c\})\times$ continuous uniform$([1\ 5],\{t\}))$
kcf	continuous uniform$([5\ 10],\{k\})$
cap^{train}	$\dfrac{\sum_{j,c,t}(vol_j\times dem_{jct})}{K}\times$ continuous uniform$([0.9\ 1.2])$
vol_j	continuous uniform$([4\ 8],\{j\}))$
φ	$\dfrac{\sum_{\hat{c}c}tm_{cc}^{travel}}{K}\times$ continuous uniform$([0.5\ 1.5])$
dd_{jct}	Ceil$(0.01\times dis_{ij}\times$continuous uniform$([1\ 3],\{i,i\}))$
$\cos t_k^{train}$	continuous uniform$([500\ 1500],\{k\})\times kcf\{k\}$
η	0.01
$\cos t_j^{price}$	continuous uniform$([1\ 4],\{j\})$
A	Ceil$([\sum_k c_k^{train}+c^{trs}\times\dfrac{\sum_{i,j}dis_{\hat{c}c}^{train}}{N\times K}+\sum_{i,p}(c_{ip}^{man}\times\dfrac{\sum_i dem_{jct}}{N})]\times$continuous uniform$([0.5\ 0.8]))$

Table 2. *Comparison between the performance of GAMS and hybrid GA-SA (optimality and run time) for small sized problem.*

Problem number	Comparison							
	GAMS		GA-SA		$\dfrac{(GASA-GAMS)}{GAMS}$	$(\dfrac{GASA}{GAMS})$	GAMS error message	GA-SA algorithm Stop condition
	Objective value	Run time	Objective value	Run time	Objective value	Run time		
P01	1771000	19.232	1771000	4.323	0	0.224782	-	-
P02	2941900	21.221	2941900	4.213	0	0.19853	-	-
P03	2019900	23.932	2071180	4.196	0.025387	0.221635	-	-
P04	11315000	28.382	12092540	4.983	0.068718	0.175569	-	-
P05	23879000	35.24	24762360	5.893	0.036993	0.167225	-	-
P06	27042000	36.291	28374070	5.362	0.049259	0.156368	-	-
P07	34257000	45.234	36677180	7.291	0.070648	0.161184	-	-
P08	43188000	49.933	46105930	6.053	0.067563	0.144349	-	-
min	1771000	18.932	1771000	4.196	0.070648	0.144349	-	-
mean	18523100	27.16278	19349520	5.28925	0.039821	0.171449	-	-
max	43188000	45.234	46105930	7.291	0.067563	0.224782	-	-

Table 3. *Comparison between the performance of GAMS and hybrid GA-SA (optimality and run time) for medium sized problem.*

Problem number	Comparison							
	GAMS		GA-SA		$\dfrac{(GASA-GAMS)}{GAMS}$	$(\dfrac{GASA}{GAMS})$		
	Objective value	Run time	Objective value	Run time	Objective value	Run time	GAMS error message	GA-SA algorithm Stop condition
P09	97787000	64.242	105022500	10.7762	0.073992	0.167744	-	-
P10	205352700	68.485	221185350	11.0938	0.0771	0.161989	-	-
P11	369305700	71.854	403619480	11.731	0.092914	0.163262	-	-
P12	1065001800	82.734	1168635560	14.8388	0.097309	0.179356	-	-
P13	3070961300	119.537	3295015400	17.2633	0.072959	0.144418	-	-
P14	5646683700	131.772	6108022200	17.9937	0.081701	0.136552	-	-
P15	16160102500	141.293	17931197800	16.2919	0.109597	0.115306	-	-
P16	28585651800	149.2131	31591748100	13.8741	0.105161	0.092982	-	-
Min	97787000	64.242	105022500	10.7762	0.109597	0.092982	-	-
Mean	6900105813	103.6413	7603055799	14.23285	0.088842	0.145201	-	-
Max	28585651800	149.2131	31591748100	17.9937	0.072959	0.179356	-	-

Table 4. Comparison between the performance of GAMS and hybrid GA-SA (optimality and run time) for large sized problem.

Problem number	Comparison							
	GAMS		GA-SA		$\frac{(GASA-GAMS)}{GAMS}$	$(\frac{GASA}{GAMS})$	GAMS error message	GASA algorithm Stop condition
	Objective value	Run time	Objective value	Run time	Objective value	Run time		
P17	5.49217E+11	1000	5.07519E+11	1000	-0,08216	1	Resource limit exceeded	Time limit exceeded
P18	1.63698E+12	1000	1.45287E+12	1000	-0,12672	1	Resource limit exceeded	Time limit exceeded
P19	-	-	4.75602E+12	1000	-	-	Out of memory	Time limit exceeded
min	5.49217E+11	1000	5.07519E+11	1000	-0,12672	1	-	-
mean	7.28732E+11	666.67	22.3882E+11	1000	-0.06962	0.66	-	-
max	1.63698E+12	1000	4.75602E+12	1000	-0,08216	1	-	-

6. Conclusions

A new model for the integration of production and distribution scheduling in supply chain management has been developed in this paper which minimizes the costs of transportation and tardiness of delivery a product. We also considered rail transportation to move deteriorating items. A hybrid genetic algorithm-simulated annealing is designed to solve the model for the large-sized problem instances of this mixed integer programming problem. Some small sized and medium-sized test problems have been solved. The results obtain from the hybrid GA-SA satisfactorily compared with results obtained from GAMS optimization software. Some other large-sized test problems have also been solved using the propose hybrid algorithm.

References

[1] Beamon, B. M. (1998). Supply chain design and analysis: Models and methods. *International journal of production economics*, 55 (3), 281-294.

[2] Fahimnia, B., Farahani, R. Z., Marian, R., & Luong, L. (2013). A review and critique on integrated production–distribution planning models and techniques. *Journal of Manufacturing Systems*, 32 (1), 1-19.

[3] Ghiami, Y., & Williams, T. (2015). A two-echelon production-inventory model for deteriorating items with multiple buyers. *International Journal of Production Economics,* 159, 233-240.

[4] Goldberg, D. E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Publishing Company, Reading, MA.

[5] Hajiaghaei-Keshteli, M., Aminnayeri, M., & Ghomi, S. F. (2014). Integrated scheduling of production and rail transportation. *Computers & Industrial Engineering,* 74, 240-256.

[6] Hajiaghaei-Keshteli, M., & Aminnayeri, M. (2014). Solving the integrated scheduling of production and rail transportation problem by Keshtel algorithm. *Applied Soft Computing,* 25, 184-203.

[7] Holland, J. H. (1975). Adaption in natural and artificial systems. Ann Arbor MI: The University of Michigan Press.

[8] Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European journal of operational research,* 88 (1), 165-181.

[9] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science,* 220 (4598), 671-680.

[10] Leung, T. W., Yung, C. H., & Troutt, M. D. (2001). Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. *Computers & industrial engineering,* 40 (3), 201-214.

[11] Lundy, M., & Mees, A. (1986). Convergence of an annealing algorithm. *Mathematical programming,* 34 (1), 111-124.

[12] Maihami, R., & Karimi, B. (2014). Optimizing the pricing and replenishment policy for non-instantaneous deteriorating items with stochastic demand and promotional efforts. *Computers & Operations Research,* 51, 302-312.

[13] Pasandideh, S. H. R., Niaki, S. T. A., & Asadi, K. (2015). Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: NSGA-II and NRGA. *Information Sciences,* 292, 57-74.

[14] Priyan, S., & Uthayakumar, R. (2014). Two-echelon multi-product multi-constraint product returns inventory model with permissible delay in payments and variable lead time. *Journal of Manufacturing Systems.*

[15] Saracoglu, I., Topaloglu, S., & Keskinturk, T. (2014). A genetic algorithm approach for multi-product multi-period continuous review inventory models. *Expert Systems with Applications,* 41 (18), 8189-8202.

[16] Yaghini, M., & Akhavan, R. (2012). Multicommodity network design problem in rail freight transportation planning. *Procedia-Social and Behavioral Sciences,* 43, 728-739.

[17] Zhang, J., Liu, G., Zhang, Q., & Bai, Z. (2015). Coordinating a supply chain for deteriorating items with a revenue sharing and cooperative investment contract. *Omega,* 56, 37-49.