A Hybrid Methodology based on heuristic algorithms for a production distribution system with routing decisions

Mohamed Bensakhria*, Samir Abdelhamid*

* University of Batna 2, Fesdis, Algeria

Abstract: In this paper, we address the integration of a two-level supply chain with multiple items. This two-level production-distribution system features a capacitated production facility supplying several retailers located in the same region. If production does occur, this process incurs a fixed setup cost and unit production costs. Besides, deliveries are made from the plant to the retailers by a limited number of capacitated vehicles, routing costs incurred. This work aims to implement a minimization solution that reduces the total costs in both the production facility and retailers. The methodology adopted based on a hybrid heuristic, greedy and genetic algorithm uses strong formulation to provide a suitable solution of a guaranteed quality that is as good or better than those provided by the MIP optimizer. The results demonstrate that the proposed heuristics are effective and performs impressively in terms of computational efficiency and solution quality.

Keywords: optimization, production-distribution systems, routing decision, mathematical modeling, genetic algorithms.

1. Introduction

The coordination of different parties in a supply chain has become more important in current, competitive markets. Therefore, integrating and utilizing production facilities, manufacturers, distribution centers, and retailers; has a significant influence on many firms’ financial performance, according to Geoffrion et al. (1995) and Thomas et al. (1996).

Over the recent years, modeling and simulation of production routing problems (PRP) have been well developed; given the huge role it plays in the analysis of the discrete behavior of this type of system, this later combines lot-sizing decisions, inventory management, and routing. In the PRP within each period, the production-facility must
determine whether or not to make the product and decide the corresponding quantity. They were considering production setup cost and variable cost, transportation, and routing costs. According to recent studies, integrated supply chain planning problems consider the optimization of stock and transport costs separately which is significant and cannot be neglected, Jie et al. (2005) a supply chain is composed of layers. Deliveries made between each pair of layers. Each pair represents one level of the distribution network; transportation is considered the building block of any supply chain—many shipment policies proposed in the literature for this problem. For instance, Infante et al. (2009) a current trend in the application of freight distribution systems involving two or more levels where at first freight travels using ships and then is loaded on trucks to distribute it over the warehouses or retailers. The problem we address here has an essential similarity with the two-level routing problems shown in Figure 1, a class of problems that study how to optimally route products in two-level distribution systems, in contrast to location decisions. In this research, a two-level distribution network system involving three parties: retailer, means of transportation, and production-facility. This latter produces a single item product and provides it to a fleet a vehicle, while the vehicles deliver the product to its retailers. The retailers' demand for a particular product arrives dynamic and deterministic.

![Figure 1. Structure of two-level problems](source: Author)

Hence, the retailer's inventory depletes over a finite discrete horizon of periods; this process drives fixed and variable costs. The distribution uses a set of identical fleets of vehicles subject to limited capacity constraints from the production capacity, number of vehicles, and routing costs. In contrast, the PRP differs from the classical VRP (Vehicle routing problem). It deals with the mixed fleet or as the heterogeneous fleet of vehicles having different capacities and costs (Xu & Jiang 2014).

The remainder of this paper is organized as follows: Section 2 presents the notation used throughout the paper and introduces the PRP literature concept. In contrast, the problem and the mathematical formulation with its properties as the proposed model are hard to solve, especially in large problem sizes presented in section 3. Section 4, a two-phase algorithm, is presented to solve the proposed model and satisfy our
constraints. In section 5, we provide experimentation and results. Finally, Section 6 provides the conclusions drawn and future research lines.

2. State of the art

PRP tackles the decisions of the developments in integrated production, inventory, distribution, and routing in manufacturing environments. The benefits of the coordination in the PRP were first discussed by Chandra et al. (1994). They provided comprehensive reviews on the general subject. An integrated production routing problem is the problem of simultaneously finding the decision variables from different functions that have traditionally been optimized independently. The four main groups of decisions to be taken into account consist of (i) quantities to be produced in supplier at each planning period, (ii) quantities to be delivered to retailers, (iii) inventory in retailers at the end of each planning period, (iv) vehicles to be used, and retailers to be served by each vehicle.

In the research literature, these four decisions can be found in Min and Zhou (2002), Bhatnagar et al. (1993), Ahmadizar et al. (2015) and Cárdenas et al. (2019) developed a heuristic algorithm to solve the selective and periodic inventory routing problem. They proposed a reduced and optimized approach with the introduction of a new inequality. Iassinovskaia et al. (2017) addressed a pickup and delivery inventory-routing problem within time windows (PDIRPTW) over a planning horizon. Regarding a large number of studies in this area, they used different tools and techniques in past publications for the optimization of the PRP problem. Mathematical techniques which have been widely used in the past include heuristic procedures. Several metaheuristics, such as Boudia (2009), using the memetic algorithm with dynamic population management for an integrated production-distribution problem. Patiño et al. (2016) developed a mixed-integer programming model and a clustering methodology for waste collection and transportation processes. The main disadvantage of this approach is that it does not consider the production capacity, which is a crucial constraint in the PRP systems. Chang et al. (2011) discussed a serial production system with finite storage space that follows a base-stock policy. They developed a phase-type approximation to estimate the effective arrival rate of customers to determine the optimal base-stock level. Bard and Nananukul (2009) presented a comparative analysis of a series of heuristics for an inventory routing problem (IRP) using a branch-and-price framework. Golden et al. (1984) were the first to investigate the interrelated problem of inventory allocation and vehicle routing. They developed a heuristic for designing an integrated delivery planning system aimed at comparing the distribution rule used by the company with their approach.

Recently, new forms of relationships adopted in the supply chain have. One of these is called vendor managed inventory (VMI), Archetti et al. (2007) have introduced a particular policy, known as deterministic order up to level policy, the problem is to evaluate quantities, and routing decision for each period, they present a mixed-integer linear programming model with new valid inequalities to strengthen the linear relaxation. Chitsaz et al. (2019) presented a general model for the assembly routing problem (ARP). The problem formulated as a mixed-integer linear program, and they
proposed three-phase decomposition metaheuristics that rely on the iterative solution of different subproblems. We refer to Axsater (2001) and Rafie-Majd et al. (2018) for application of the VMI policy to systems with stochastic demand. When the routing aspect includes the production aspect disregarded, the problem becomes a particular case of the inventory routing problem (IRP). Recent reviews of the rich literature about the IRP can be found in Bell et al. (1983); they proposed a lagrangian relaxation algorithm to solve mixed-integer programs in analyzing the route selection model. Miranda et al. (2019) formulated a MIP and proposed several relax-and-fix (RF) heuristics to solve a particular PRP in small furniture companies. They considered a scenario in which the manufacturer has one production line and only one vehicle that may perform multiple trips over the planning horizon.

Jie et al. (2005) presented a production and inventory planning models with a piecewise linear transportation cost function where the supplier has the option delivery with a truckload (TL) or less than truckload (LTL) transportation, an efficient dynamic programming algorithm is developed through the use of Monge matrices. Lee et al. (2003) also examined a model with multi-mode shipments. The outbound shipments have different lead times, and multiple items are delivered to multiple customers using dynamic programming algorithms to solve the problem.

In the more general of lot-sizing problems (LSP), a piecewise linear transportation cost function considered by Rizk et al. (2006). A Lagrangian relaxation method used to decompose the problem into a set of simple subproblems. As mentioned in Federgruen et al. (1999) the case of uncaptacitated production and vehicles, which is mainly related to two well-known problems: the one-warehouse-multi retailer (OWMR) and the production routing PRP, they proposed a general approach to develop an efficient heuristic for (OWMR) model to satisfy all the optimality criteria. Melo and Wolsey (2012) discussed a two-level supply chain with multiple items, capacitated production, and vehicles. They introduced a variety of formulations with hybrid heuristic in which production sites and client areas are identical, and products can be transferred between production sites.

Concerning inventory models with transportation costs, Bilgen and Günther (2009) presented an integrated production scheduling and truck routing model for a supply chain. Due to the complexity of the problem, the VRP plays a central role in physical distribution and logistics. The VRP generalization determines the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers. Indeed, Swenseth and Godfrey (2002) account that more than 50% of the total logistics costs can be related to transportation. Other researchers, including Sun et al. (2015), Miranda et al. (2019) and Wang et al. (2015), stress the importance of considering and combining transportation decisions into PRP models. Other related problems are associated with VRP, such as the Heterogeneous Fleet Vehicle Routing Problem (HVRP), Xu and Jiang (2014) focused on solving the Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows (MDHVRPTW) via an Improved Variable Neighborhood Search (IVNS) metaheuristic. Tarantilis et al. (2001) studied an extension of the classical VRP, named (HFFVRP) Heterogeneous Fixed Fleet Vehicle Routing Problem based on threshold accepting based algorithm aiming to find effective delivery schedules for one of the biggest dairy companies in Greece. Gong
et al. (2010) built a model for a multi-objective perishable food distribution problem that includes shelf life lost with time window default cost and the vehicle routing problem, the approach based on a two-generation ant colony algorithm and a comparative experiment.

In this paper, the problem mathematically formulated by a MILP model. Due to the complexity of the problem which shown to be NP-hard by Arkin et al. (1989), two well-known heuristics available in the literature of PRP (Genetic Algorithm (GA) and Greedy Algorithm (GA)), are first adapted to (LSP) and (VRP). To more effectively solve larger problems, a two-phase algorithm also presented to solve the proposed approach. The algorithm is greedy, considering at each step, several alternatives generated, and the best one is selected. Different numerical experiments performed. The results show the approach and the heuristics are adequate for the problem.

3. Problem description

In this section, we begin to explore this problem by recognizing the classified points. We assume a two-stage supply chain that faces a fixed nonnegative deterministic demand $d_{rt}$ for a single product that must be satisfied over time. A set of $r = \{1\ldots R\}$ retailers geographically dispersed around the production facility where a unique product is stored. A single product delivered to retailers over a finite set of time $t = \{1\ldots T\}$, backorders are not permitted, the excessive product can be stored in both, denoting by $h_p, h_r$ the unit inventory cost at the production facility and the retailers, respectively.

The quantity produced at the production facility in each period restricted to $P_{\text{max}}$, the storage capacity at the supplier and retailer is limited to $C_p, C_r$, respectively. There is a resulting fixed production setup cost $F_{pc}$, and a variable setup cost $\nu_{pc}$. A fleet of $k = \{1\ldots K\}$ homogeneous vehicles make deliveries from the production facility to retailers with capacity $Q$, and fixed transportation cost $C_{ij}$ when the link between two pairs $i$ and $j$ traversed. Each vehicle route visits all retailers that are served at the same time. Routing decisions must be solved each period to guarantee a feasible route and determine which customers to visit and how much product to deliver.

The two-stage supply chain design requires two critical decisions to be made. The first set of decisions involves a construction delivery schedule that minimizes production, inventory, and routing costs simultaneously while ensuring that each customer’s demand is met and the inventory limits at the production facility and retailers are satisfied. In the second set of decisions, the model must determine the quantity delivered to each retailer, subject to product availability, and total vehicle capacity. From the comparative study of different extensions of the PRP problems and our objectives, the problem formulated as a mixed-integer linear programming (MILP) with a multi-heuristics-based approach.

Indexes

$t = 1\ldots T$ : set of periods
$r = 1\ldots R$ : set of retailers
\( k = 1 \ldots K \) : set of vehicles  
\( d_{rt} \): demand of retailer \( r \) in period \( t \)  
\( C_r \): storage capacity of retailer \( r \)  
\( C_p \): storage capacity of production facility  
\( P_{\text{max}} \): production capacity  
\( h_r \): inventory holding cost of retailer \( r \)  
\( h_p \): inventory holding cost of production facility  
\( Q \): vehicle capacity  
\( F_{pc} \): fixed setup cost  
\( V_{pc} \): variable setup cost  
\( C_{ij} \): fixed transportation cost

**Decision variables**

- \( D_{rkt} \): The quantity to be delivered to retailer \( r \) using vehicle \( k \) in period \( t \)  
- \( I_p^t \): Inventory level at production facility in planning period  
- \( I_r^t \): Inventory level at retailer \( r \) in planning period \( t \)  
- \( Z_t \): Production quantity in planning period \( t \)  
- \( Y_t \): Binary variable equal 1 if there is production on planning period \( t \), otherwise 0  
- \( X_{rkt} \): Binary variable equal 1 if vehicle \( k \) is used in planning period \( t \) to serve retailer \( r \) in period \( t \), otherwise 0  
- \( U_{rkt} \): Binary variable equal 1 if retailer \( r \) is served by vehicle \( k \) in planning period \( t \), otherwise 0

\[
\text{Min} = \sum_{t=1}^{T} \left( F_{pc} \cdot Y_t + V_{pc} \cdot Z_t + h_p \cdot I_p^t + \sum_r h_r \cdot I_r^t + \sum_r \sum_k C_{ij} \cdot X_{rkt} \right)
\]

(1)

\[
I_p^t = I_p^{t-1} + Z_t - \sum_r \sum_k D_{rkt} \quad \forall r \in R, k \in K, t \in T
\]

(2)

\[
I_r^t = I_r^{t-1} + \sum_k D_{rkt} - d_{rt} \quad \forall r \in R, k \in K, t \in T
\]

(3)

\[
I_p^t \leq C_p \quad \forall t \in T
\]

(4)

\[
I_r^t \leq C_r \quad \forall t \in T
\]

(5)

\[
Z_t \leq P_{\text{max}} \cdot Y_t \quad \forall t \in T
\]

(6)

\[
D_{rkt} \leq Q \cdot U_{rkt} \quad \forall r \in R, k \in K, t \in T
\]

(7)

\[
\sum_r D_{rkt} \leq Q \quad \forall r \in R, k \in K, t \in T
\]

(8)

\[
\sum_k U_{rkt} \leq 1 \quad \forall k \in K, t \in T
\]

(9)

\[
\sum_r U_{rkt} \leq 1 \quad \forall r \in R, t \in T
\]

(10)

\[
\sum_r U_{rkt} \leq Q \cdot X_{rkt} \quad \forall k \in K, t \in T
\]

(11)

\[
X_{rkt}, Y_t, U_{rkt} \in \{0,1\} \quad \forall r \in R, k \in K, t \in T
\]

(12)

\[
Z_t, I_p^t, I_r^t, D_{rkt} : \geq 0 \quad \forall r \in R, k \in K, t \in T
\]

(13)

In this formulation, the objective function (1) minimizes the total costs of production, inventory, and transportations. Constraints (2) and (3) are the inventory flow balance at the production facility and retailers, respectively, where the inventory level at the end of period \( t \) can be obtained by adding the inventory level at the end of period \( t-1 \). Constraints (4) and (5) give an upper limit on the inventory levels at production facility and retailers. Constraints (6) limit the maximum inventory at the production facility. Constraints (7) link the continuous variable \( D_{rkt} \) to the binary variable, which means if there is a delivery to a retailer in the period \( t \); otherwise, it must be 1. Constraints (8) represent the capacity restriction of the vehicle. Constraints (9)
A Hybrid Methodology based on heuristic algorithms for a production distribution system with routing decision

Title of the paper

guarantee that each retailer is visited at most once in a given period. Similarly, constraints (10) assure the vehicle capacity and routing restrictions. Constraints (11) determine the number of visits in period t. Finally, constraints (12) and (13) represent the variable definitions and assure the nonnegativity restrictions on the decision variables.

As shown in the proposed formulation, the PRP is a complicated combinatorial optimization problem containing a large number of binary variables. Initial attempts to solve instances of this size with CPLEX 8.1 were not encouraging, and solving the LP relaxation of the PRP formulation using the quality of the lower bound is generally very poor. Duc and Nananukul (2020) studied the LP relaxation of the PRP formulation, which is not useful in solving solutions in exact algorithms. Hence, to solve such a complex problem is to split into two problems: the master problem and the subproblem, which are typically easier to solve than the original problem. Note that the delivery amounts and the number of binary variables is the key decision variables whose values can be easily used to determine the rest of the decision variables, including vehicle tours.

4. Proposed method

Due to the complexity of our problem and for efficiency reasons, a solution procedure is based on a two-phase algorithm, decomposition, and integration. The first part determines the shortest path constraints for the production facility submodel, which contains all the original constraints. Resulting solutions to this phase is always feasible but not optimal and thus gives an upper bound solution to our problem. The goal of this decomposition is to form a production plan and delivery schedule that minimize the sum of all costs while ensuring that each customer's demand is met each period.

Resulting solutions determines production quantities, shipped amounts to each retailer, and each retailer that must be visited every period. Fixing the routing decisions in the underlying stage characterizes the number of visited retailers and the transported amount. Therefore, in the following stage, we propose a heuristic to solve the routing problem in the distribution submodel. In the next cycle, the second submodel's results are considered a contribution to readjust the routing choices for the following emphasis. The methodology proceeds with this iterative solve feedback solve until a stopping criterion is accomplished or achieve a fixed number of redundancies. Algorithm 1 illustrates the two-phase procedure.

ALGORITHM 1: Iterative method

Begin
Step 1: Estimate demand and production quantity for each retailer that needs delivery in period t
Step 2: Solve production problem using Greedy algorithm
Step 3: Solve the routing problem using Genetic algorithm
Step 4: Calculate sol and set best sol = sol
Repeat
   Repeat
Step 5: Update demand and production quantity for each retailer
Step 6: Run diversification mechanism
Step 7: Solve production problem using retailers’ updated costs using Greedy algorithm
Step 8: Solve routing problem using Genetic algorithm
Step 9: Update solution if necessary
Until a stopping criterion is met
Step 10: Calculate the solution

If (solution < best solution)
Set best solution = solution
End if
Until a stopping criterion is met

End.

The production facility can be modeled as an integer linear programming. Replenishment demands from the retailers arrive according to a deterministic approach and wait to be processed. As previously mentioned, the lot-sizing phase determines quantities to be produced, quantities to be delivered to retailers, and retailers to be serviced by each vehicle over a planning time horizon. The idea behind this procedure is to try to accumulate the retailer's demand and the vehicle's capacity, using the retailer's orders \( D_{rt} \) since in this model we assume that the starting inventory is zero, and we require all demand to be met, we do not allow demand in any period before period \( t \). The production facility submodel shown in relations (14) to (19):

\[
\begin{align*}
\text{Min} &= \sum_{t=1}^{T} (F_{pc} \cdot y_T + V_{pc} \cdot Z_t + h_p \cdot I_p^t) \\
I_p^t &= I_p^{t-1} + Z_t - D_{rt} \quad \forall \ r \in R, k \in K, t \in T \\
I_p^t &\leq C_p \quad \forall \ t \in T \\
Z_t &\leq P_{max} \cdot Y_t \quad \forall \ t \in T \\
Y_t &\in \{0,1\} \quad \forall \ r \in R, k \in K, t \in T \\
Z_t, I_p^t &\geq 0 \quad \forall \ t \in T 
\end{align*}
\]

At this point, the algorithm continues to the second part that requires all the routing decisions, which incorporate retailers served by each vehicle. Given that the underlying stage dispenses retailers to vehicles in addition to vehicles limit constraints. Relations (20) to (30) introduce the distribution submodel.

\[
\begin{align*}
\text{Min} &= \sum_{t=1}^{T} \left( \sum_r \sum_k C_{ij} \cdot X_{rkt} + \sum_r h_r \cdot I_r^t \right) \\
I_p^t &= I_p^{t-1} + Z_t - \sum_r \sum_k D_{rkt} \quad \forall \ r \in R, k \in K, t \in T \\
I_r^t &= I_r^{t-1} + \sum_k D_{rkt} - d_{rt} \quad \forall \ r \in R, k \in K, t \in T \\
I_r^t &\leq C_r \quad \forall \ t \in T \\
D_{rkt} &\leq Q \quad \forall \ r \in R, k \in K, t \in T \\
\sum_r D_{rkt} &\leq Q \quad \forall \ r \in R, k \in K, t \in T \\
\sum_k U_{rkt} &\leq 1 \quad \forall \ k \in K, t \in T \\
\sum_r U_{rkt} &\leq 1 \quad \forall \ r \in R, t \in T 
\end{align*}
\]
Title of the paper A Hybrid Methodology based on heuristic algorithms for a production distribution system with routing decision

\[ \sum_r U_{rkt} \leq Q \times X_{rkt} \forall k \in K, t \in T \]  
(28)

\[ X_{rkt}, U_{rkt} \in \{0, 1\} \forall r \in R, k \in K, t \in T \]  
(29)

\[ Z_t, I_{r}^p, I_{t}^r, D_{rkt} \geq 0 \forall r \in R, k \in K, t \in T \]  
(30)

In our first formulation, we developed a greedy search algorithm to generate feasible solutions using the production submodel; the initial tests found that these heuristic solutions are unlikely to perform well. Even when they can be adapted, they may not be able to find optimal solutions in a reasonable time. To extract a feasible solution, we use an adaptation to include genetic algorithms. Consequently, the question we ask is how to adapt two different heuristics effectively.

4.1 Genetic algorithm proposed for the minimization of the total transport cost

The genetic algorithm (GA) has shown to be a highly effective and efficient tool in solving complex optimization problems. Some of their successful applications in the optimization of supply chain models have been proposed in literature Gen and Syarif (2005). The genetic algorithm emulates the process of natural selection. Indeed, in order to solve optimization problems, genetic algorithms must pass through a few steps starting by generating a random population of individuals. Then the three operations: crossover, mutation, and evaluation are repeated for all population m. Each potential solution portrayed into a finite length array called a chromosome.

4.2 Proposed genetic algorithm

Binary encoding gives many possible chromosomes representing the solutions; in the first phase, chromosomes are created for optimal size. We describe the routing phase, which corresponds to calculating vehicle routes at each period. A chromosome is a sequence of nodes indicating the order in which vehicles must visit all retailers. In this case, we consider a set of retailers served by a fleet of identical vehicles in planning periods; we set a two-dimensional matrix that each cell contains the retailer’s demand in a given period. An example of an encoding structure with four retailers, two vehicles, and five planning periods is shown in table 1

**Table 1. Parameters of encoding structure**

|       | k1 | k2 |
|-------|----|----|
| r1    |    |    |
| r2    |    |    |
| r3    |    |    |
| r4    |    |    |
| r1    |    |    |
| r2    |    |    |
| r3    |    |    |
| r4    |    |    |

*Source: Author*

The binary variables Urkt that imply if a vehicle k is used in planning period t to serve a retailer r is the main issue to identify the other decision variables. To quickly converge to global optimum, we used the integer rule overall deliveries. The choice of setting the quantities integers is crucial because it can make the search process more
straightforward and faster, given the fact that all the values of the retailers' demand, production quantities, and vehicles capacities are integers. More specifically, in this case, it is presumed without loss of generality. We repeat the above procedure until a stopping criterion is achieved. The details of the basic genetic algorithm are given below in algorithm 2.

**ALGORITHM 2: Genetic algorithm**

Begin
Step 1: Population size (m)
Step 2: Number of generations (n)
Step 3: Create a delivery with a random order of retailers
Step 4: Do this for m times to create a population
Step 5: Select the best delivery from the population
Step 6: Set the initial feasible solution as local best solution and global best solution
Genetic algorithm
Run for n times
    Loop1
    Run for m times
        Loop2
        Select a random set of delivery from the population
        Get the fittest and return
        //Crossover
        Parent1 = roulette wheel selection ()
        Parent2 = roulette wheel selection ()
        Child = Crossover (Parent1, Parent2)
        //Mutation
        Swap random two retailers in the child
    endLoop2
    //New population is created
    Calculate the fitness value for each individual
    Sort the population by their fitness values
    Update local best Solution and global best Solution
endLoop1
Final local solution = local best Solution
Final global solution = global best Solution
End

**4.3 Genetic operators**

Genetic variation is a necessity for the process of evolution. However, when producing the new population using genetic operators, there is a chance of losing the best-found chromosomes. Given the model constraints the following assumptions must be respected under any circumstances: production capacity, storage capacity and vehicles delivery one tour at most each time period. Throughout the algorithm instructions, if any of the above conditions were violated, we introduce in the fitness an overproduction penalty for the production capacity, the other rules we consider an updating algorithm.
The mechanism for generating the initial population allows the production of individuals’ population as the basis for future generations. The choice of the initial population is important because it can make more or less fast the convergence to the global optimum.

4.3.1 Selection and crossover

The genetic algorithm first part initiates with selection, in which we select individuals for crossover according to their relative fitness. In this way, we implement the selection step in practice by creating a mating pool for the next generation. The choice of a selection technic in the suggested genetic algorithm is the roulette wheel operator, which selects the initial population to create new children. The probability of choosing an individual depends directly on their fitness.

In the next step, the crossover occurs when two members of a population (chromosomes) are selected for reproduction and is usually based upon their relative fitness in solving the problem. Two parents P1 and P2 are chosen randomly; we use the order crossover (OX) with slight modification. We begin by choosing two crossover points then applying a sliding motion to complete the excluded holes by sending the designated positions. It copies the permutation elements between the crossover points from the cut string directly to the offspring, placing them in the same absolute position. We illustrate an example of offspring generation in Figure 2.

**Figure 2.** Order crossover operator

| Parent 1 | Parent 2 |
|----------|----------|
| 1 4 6 2 8 0 3 7 9 5 8 | 7 8 5 4 3 0 6 1 2 9 3 |

| Child 1 | Child 2 |
|---------|---------|
| 3 6 1 2 8 0 3 7 9 8 4 | 8 7 5 4 3 0 6 1 9 5 2 |

*Source: Author*

4.3.2 Mutation

The mutation may be defined as a small perturbation to prevent the genetic algorithm converging to a local minimum; we opted to use the operator left to right flip for the mutation process. An example of a mutation consists of choosing randomly two genes from the chromosome, and exchanging their values Figure 3.
As noted, to escape such a point, we implemented a diversification mechanism. The rule we adopt it captures, for each retailer, the complementarities with the other retailers served on the same route. Our principle of correction is to start by calculating the total sum of the quantity transported by vehicles. After that, we have to browse each group gene by gene, and if it is higher than the maximum available vehicle capacity, we must search in the next sequence of group’s nodes which succeed this position of excess capacity, a retailer who has not served and its insertion does not cause an overload of vehicle we insert it in the next position.

4.4 Repair algorithm

Explicitly that our decomposition approach into two-phase iterative heuristic, then solving each heuristic separately, may cause the search to be stuck in a local optimum. More particularly, throughout the lot-sized phase, we noticed an increased delivery considering the quantities determined in the precedent phase. Therefore, it is necessary to employ a perturbation strategy to diversify the search. Three mechanisms have been developed: production planning, determining delivery quantities, and assigning deliveries to vehicles.

- If the retailers demand overcome the production capacity, we shift the additional amount to the next time period \( t+1 \), after the first phase completed the new quantity for period \( t+1 \) defined by \( FD_{rt} \) the feasibility of this solution is verified by the constraint (31)

\[
\sum_{t=1}^{t} FD_{rt} \leq t \times P_{\text{max}} \forall t \in T, r \in R
\]  

(31)

- Delivery quantity, once the periods of delivery for each retailer are known according to the fitness function, the quantities calculated using the zero-inventory property (32), which consists of ordering only when there is a zero stock at the beginning of period, so as to satisfy the demand of successive periods.

If \( U_{rkt} = 0 \), then \( D_{rkt} = 0 \). For \( t_1 \) and \( t_2 \), \( U_{rkt_1} = 1 \), \( U_{rkt_2} = 1 \), and \( U_{rkt} = 0 \):

\[
t_1 < t < t_2, D_{rkt} = \sum_{t_1}^{t_2-1} d_{rt}
\]  

(32)

- For the assigning delivery quantities to vehicles, the goal is to determine the number of vehicles required and if it exceeds the number of vehicles, the
number of feasible solutions decreased and must be repaired by introducing a penalty.

The temporary solution obtained is compared with the current solution to decide whether there is an improvement. The goal is to employ a strategy of accepting non-improving solutions by searching a new promising region that was not explored recently is an effective strategy. This mechanism favors the ejection of retailers from periods where the number of visits is high to periods where the number of visits is low. An illustrative example of the repair algorithm is presented in algorithm 3.

Algorithm 3: Update of the visiting cost

Begin
Step 1: Creation of the initial population
While Max number of generations not reached and stopping criterion not achieved Do
   Step 2: Creation of the intermediate population
   Step 3: update the repairing mechanism
   Step 4: Creation of the second intermediate population
   Step 5: Calculate the fitness values for each individual
   Step 6: Sort the population according to the minimum fitness values
   Step 7: Copying the non dominated solutions;
   Step 8: Increment the generation number;
End While
End

The repair algorithm includes two phases: the primary and restart phases; the main phase stops after a maximum number of iterations. The first, the third, the fifth … chromosomes of the population are kept, and new randomly generated chromosomes replace the rest. The main phase is repeated but with a higher local search probability. After that, the restart procedure is implemented for n times. It is worth noting that the proposed partial replacement procedure in the restart phase can reserve good chromosomes, and increase the diversity of the population because any two adjacent chromosomes in the sorted populations are often too close to cloning after many iterations of each phase.

5. Experimental results

In this section, the effectiveness of this work is discussed with regard to objective functions, CPU time, average gap and optimal solutions. Our algorithms are implemented in Matlab environment. We used IBM ILOG CPLEX 12.6 with the default settings to solve the proposed problem. All experiments were performed on an Intel Core 2Duo processor, with 4096Ko of memory cache, 4GB of RAM and a frequency of 3.0 GHz using the Windows 7 operating system.

5.1 Test instances

We report computational tests using two types of randomly generated instances small and large sizes. We construct data sets from existing problems from the literature,
more particularly the instances proposed in (Archetti et al., 2011). Computational tests were done on problems, which were generated by varying the following parameters, in addition to parameters of table 2:

- Number of periods (T)
- Number of retailers (R)
- Number of vehicles (K)
- Production capacity (P\textsubscript{max})
- Vehicle capacity (Q)

(i) Demand quantities were generated from a uniform distribution \( \text{drt} = [5, 25] \)
(ii) Variable production setup costs were generated from a uniform distribution \( \text{Vpc} = [50, 100] \).
(iii) Fixed production setup costs were generated from a uniform distribution \( \text{Fpc} = [500, 1000] \).
(iv) Inventory holding costs in production facility and retailers were generated from a uniform distribution \( h = [5, 10] \).
(v) Fixed transportation costs were generated from a uniform distribution \( \text{Ci,j} = [200, 300] \).
(vi) Vehicle capacity was set to \( 2\alpha, 1.5\beta \) (\( \alpha = \sum_{r,t} \text{td}_{r,t} / R \times T \)).
(vii) Production capacity was set to \( 2\alpha, 1.5\beta \) (\( \beta = \sum_{r,t} \text{td}_{R,t} / R \times T \)).

| Table 2. Parameters of the test problems |
|------------------------------------------|
| Small size | Large size |
| Problem number size (R×T×K) | Problem number size (R×T×K) |
| \( R \) | 4, 6, 8, 10 | 30, 40, 50, 60 |
| \( T \) | 5, 10 | 10, 15 |
| \( K \) | 2, 3, 4, 5 | 10, 15, 20, 25 |

Source: Author

5.2 Computational results on the small and large sets of instances

Let us first compare the effectiveness of the iterative method proposed in this paper on the small size of instances. In the latter part, we present our best solutions and average results on 32 random test problems. A summary of the results for small size test problems is reported in Table 3. In columns 2–4 a comparison of objective functions of the IBM ILOG CPLEX 12.6, Proposed Genetic Algorithm (PGA) and modified genetic algorithm (MGA), respectively, the optimal solution or the best solution is determined by the standard solver IBM ILOG CPLEX 12.6 after one hour. A summary of the results for small instances is reported in Table 3. In columns 2–4 of the table, the objective function obtained by IBM ILOG CPLEX 12.6, the proposed genetic algorithm PGA and the number of optimal solutions found, respectively. Columns 5 and 6 show the modified genetic algorithm MGA and optimal solutions. In terms of average optimal solutions the PGA is able to produce better results at the beginning afterward could not further improve the solution quality due to the algorithm sensitivity to the number of periods (T) and the increase number of vehicles (K), whereas MGA algorithm could provide better average results for 27 out of 32 test problems (13 out of 32 for PGA algorithm) and consequently on the overall number of instances solved to optimality for PGA and MGA, 544 of 648 (84%) and 252 of 648 (39%), respectively.
Table 3. Objective functions of IBM ILOG CPLEX, PGA and MGA on small instances

| Problem | CPLEX 1h | PGA | Optimal solutions | MGA | Optimal solutions |
|---------|----------|-----|-------------------|-----|------------------|
| 4 5 2   | 23465    | 23465 | 6/6               | 23465 | 6/6              |
| 4 5 3   | 24158    | 24158 | 6/6               | 24158 | 6/6              |
| 4 5 4   | 30058    | 30158 | 4/6               | 30058 | 6/6              |
| 4 5 5   | 28522    | 28862 | 3/6               | 28522 | 6/6              |
| 4 10 2  | 34028    | 34101 | 3/6               | 34060 | 3/6              |
| 4 10 3  | 42919    | 43899 | 2/6               | 42919 | 6/6              |
| 4 10 4  | 36644    | 36644 | 6/6               | 36644 | 6/6              |
| 4 10 5  | 50899    | 52543 | 3/6               | 51210 | 5/6              |
| 6 5 2   | 61520    | 61520 | 6/6               | 61520 | 6/6              |
| 6 5 3   | 56676    | 56676 | 6/6               | 56676 | 6/6              |
| 6 5 4   | 62254    | 70214 | 0/6               | 62254 | 6/6              |
| 6 5 5   | 66846    | 67541 | 4/6               | 66846 | 6/6              |
| 6 10 2  | 55318    | 55318 | 6/6               | 55318 | 6/6              |
| 6 10 3  | 45297    | 45297 | 6/6               | 45297 | 6/6              |
| 6 10 4  | 64537    | 70546 | 1/6               | 64537 | 6/6              |
| 6 10 5  | 70759    | 80549 | 0/6               | 70987 | 4/6              |
| 8 5 2   | 65754    | 65890 | 2/6               | 65754 | 6/6              |
| 8 5 3   | 63378    | 66524 | 2/6               | 63378 | 6/6              |
| 8 5 4   | 62309    | 65231 | 2/6               | 64895 | 2/6              |
| 8 5 5   | 77375    | 77456 | 5/6               | 77375 | 6/6              |
| 8 10 2  | 73985    | 74520 | 4/6               | 74595 | 3/6              |
| 8 10 3  | 55550    | 55550 | 6/6               | 55550 | 6/6              |
| 8 10 4  | 66975    | 68652 | 3/6               | 66975 | 6/6              |
| 8 10 5  | 58334    | 60259 | 4/6               | 58334 | 6/6              |
| 10 5 2  | 74586    | 77002 | 3/6               | 74586 | 6/6              |
| 10 5 3  | 91357    | 100211| 1/6               | 91357 | 6/6              |
| 10 5 4  | 61049    | 80546 | 0/6               | 61049 | 6/6              |
| 10 5 5  | 82445    | 90546 | 1/6               | 85421 | 2/6              |
| 10 10 2 | 86542    | 100254| 0/6               | 86542 | 6/6              |
| 10 10 3 | 86524    | 99652 | 0/6               | 86524 | 6/6              |
| 10 10 4 | 87764    | 87764 | 1/6               | 87764 | 6/6              |
| 10 10 5 | 87889    | 98523 | 0/6               | 90548 | 2/6              |
| Average | 60491.13 | 64064.72 | 3.00             | 60784.94 | 5.34            |

Source: Author

Table 4 provides a comparison of computing CPU times and average gaps column 2-3 and column 4-5, respectively, with respect to the best-known solution for each heuristic and for each set of instances. Concerning the comparison between PGA and MGA it can be observed that, generally, MGA outperform PGA in the context of CPU runtime which is decreased by half. The gaps between MGA and CPLEX are less than 1% in most cases. The worst performance of the algorithm also can obtain a solution gap that is less than 4%. It is clear that the improvement algorithm has a significant effect on reducing the runtime, and consequently on the number of instances solved to optimality.
Table 4. CPU time and gap for PGA, MGA and IBM ILOG CPLEX on small instances

| Problem R T K | CPU_{PGA} | CPU_{MGA} | GAP_{PGA} | GAP_{MGA} |
|--------------|-----------|-----------|-----------|-----------|
| 4 5 2        | 0.18      | 0.09      | 0         | 0.00      |
| 4 5 3        | 0.16      | 0.10      | 0         | 0.00      |
| 4 5 4        | 0.21      | 0.12      | 0.33      | 0.00      |
| 4 5 5        | 0.31      | 0.13      | 1.18      | 0.00      |
| 4 10 2       | 3.68      | 1.25      | 0.21      | 0.09      |
| 4 10 3       | 7.69      | 3.41      | 2.23      | 0.00      |
| 4 10 4       | 22.25     | 4.01      | 0         | 0.00      |
| 4 10 5       | 32.08     | 7.18      | 3.13      | 0.61      |
| 6 5 2        | 77.96     | 32.24     | 0         | 0.00      |
| 6 5 3        | 71.46     | 33.09     | 0         | 0.00      |
| 6 5 4        | 116.62    | 64.98     | 11.34     | 0.00      |
| 6 5 5        | 143.25    | 82.65     | 1.03      | 0.00      |
| 6 10 2       | 30.52     | 18.65     | 0         | 0.00      |
| 6 10 3       | 48.93     | 22.68     | 0         | 0.00      |
| 6 10 4       | 53.78     | 32.58     | 8.52      | 0.00      |
| 6 10 5       | 61.69     | 36.65     | 12.15     | 0.32      |
| 8 5 2        | 75.18     | 38.45     | 0.21      | 0.00      |
| 8 5 3        | 188.89    | 101.65    | 4.73      | 0.00      |
| 8 5 4        | 66.32     | 55.65     | 4.48      | 3.98      |
| 8 5 5        | 202.32    | 130.65    | 0.1       | 0.00      |
| 8 10 2       | 260.36    | 195.63    | 0.72      | 0.82      |
| 8 10 3       | 114.40    | 55.69     | 0         | 0.00      |
| 8 10 4       | 181.54    | 77.58     | 2.44      | 0.00      |
| 8 10 5       | 411.38    | 180.65    | 3.19      | 0.00      |
| 10 5 2       | 245.32    | 120.36    | 3.14      | 0.00      |
| 10 5 3       | 287.69    | 99.65     | 8.84      | 0.00      |
| 10 5 4       | 589.13    | 226.32    | 24.21     | 0.00      |
| 10 5 5       | 565.02    | 190.65    | 8.95      | 3.48      |
| 10 10 2      | 503.63    | 245.32    | 13.68     | 0.00      |
| 10 10 3      | 559.54    | 342.32    | 13.17     | 0.00      |
| 10 10 4      | 600.36    | 301.32    | 0         | 0.00      |
| 10 10 5      | 625.63    | 320.32    | 10.79     | 2.94      |
| **Average**  | **192.11**| **94.44** | **4.34**  | **0.38**  |

Source: Author

CPLEX is able to obtain very good results for the smallest instances. However, for larger problem instances, it is not competitive anymore. Increasing the computation time limit to two hours (2) helps for some of the medium size problem instances, where the results in comparison to CPLEX improve considerably. Table 5 lists the results obtained for the test problems using the two versions of the heuristics. For each test problem, 5 runs were conducted, column 2 presents the objective functions obtained by IBM ILOG CPLEX after two hours of execution, column 3,4,5 and 6 show the average and the worst objective function found by the PGA and MGA.
Again, the results for large instances are similar to those observed on the objective function. The tables illustrate again how PGA determines solutions of poor quality compared to MGA, which is the most successful approach on large instances. The standard solver sometimes cannot find feasible solutions whereas both heuristics find one. Thus, computing times are largely in favor of them.

In table 6 gives the CPU time, optimality gaps and optimal solutions found, the optimality gaps of PGA vary between 2% and 33% and the ones of MGA vary between 1% and 4%, i.e. MGA is more effective for large instances. An interesting observation is that this improvement of results in most instances is at the cost of a huge increase in computation time.
6. Conclusions and future work

This paper discussed PRP’s production routing problem, which is a very complicated combinatorial problem that combines the multi-level lot-sizing problem and the vehicle routing problem. We have proposed a two-phase iterative method based on two heuristics to solve the PRP. In the first phase, a lot-sizing problem is solved using a greedy heuristic. The first phase decides the time and quantities to produce each period when to visit retailers, how much to deliver each visit. The second phase takes routing decisions into account using the genetic algorithm approach. In the GA construction phase, a randomized version of a previously developed construction heuristic is used. We developed suitable designs in the GA improvement phase, which involves two random neighborhood search mechanisms, the crossover and mutation.
Title of the paper A Hybrid Methodology based on heuristic algorithms for a production distribution system with routing decision

operations. To evaluate the performance of the method, we used several randomly generated test problems. We have tested these approaches on small and large size instances. The results show that the proposed method can provide optimal solutions. On average, modified genetic algorithm MGA outperforms the proposed genetic algorithm PGA and generates solutions within 2% of the optimal solution. In addition, it is shown that the time complexity of the proposed method depends on the linear multiplication of three parameters, the number of distribution centers, vehicles, and planning periods.

Despite the exciting results presented in this paper, this approach will not suit extensive problem instances. Our next step is to develop and test different meta-heuristics or other hybrid methods. Then, many more iterations would be possible and more elaborated diversification schemes.

References

Ahmadizar, F., Zeynivand, M., & Arkat, J. (2015). Two-level vehicle routing with cross-docking in a three-echelon supply chain: A genetic algorithm approach. Applied Mathematical Modelling, 39(22), 7065–7081. https://doi.org/10.1016/j.apm.2015.03.005

Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem. Transportation Science, 41(3), 382–391. https://doi.org/10.1287/trsc.1060.0188

Archetti, C., Bertazzi, L., Paletta, G., & Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. Computers & Operations Research, 38(12), 1731–1746. https://doi.org/10.1016/j.cor.2011.03.002

Arkin, E., Joneja, D., & Roundy, R. (1989). Computational complexity of uncapacitated multi-echelon production planning problems. Operations Research Letters, 8(2), 61–66. https://doi.org/10.1016/0167-6377(89)90001-1

Axsäter, S. (2001). A Note on Stock Replenishment and Shipment Scheduling for Vendor-Managed Inventory Systems. Management Science, 47(9), 1306–1310. https://doi.org/10.1287/mnsc.47.9.1306.9782

Bard, J. F., & Nananukul, N. (2009). Heuristics for a multiperiod inventory routing problem with production decisions. Computers & Industrial Engineering, 57(3), 713–723. https://doi.org/10.1016/j.cie.2009.01.020

Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G., & Prutzman, P. J. (1983). Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer. Interfaces, 13(6), 4–23. https://doi.org/10.1287/inte.13.6.4

Bhatnagar, R., Chandra, P., & Goyal, S. K. (1993). Models for multi-plant coordination. European Journal of Operational Research, 67(2), 141–160. https://doi.org/10.1016/0377-2217(93)90058-u

Bilgen, B., & Günther, H. O. (2009). Integrated production and distribution planning in the fast moving consumer goods industry: a block planning application. OR Spectrum, 32(4), 927–955. https://doi.org/10.1007/s00291-009-0177-4
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. https://doi.org/10.1016/j.ejor.2007.07.034

Cárdenas-Barrón, L. E., González-Velarde, J. L., Treviño-Garza, G., & Garza-Nuñez, D. (2019). Heuristic algorithm based on reduce and optimize approach for a selective and periodic inventory routing problem in a waste vegetable oil collection environment. *International Journal of Production Economics, 211*, 44–59. https://doi.org/10.1016/j.ijpe.2019.01.026

Chandra, P., & Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research, 72*(3), 503–517. https://doi.org/10.1016/0377-2217(94)90419-7

Chang, K. H., & Lu, Y. S. (2011). Inventory management in a base-stock controlled serial production system with finite storage space. *Mathematical and Computer Modelling, 54*(11–12), 2750–2759. https://doi.org/10.1016/j.mcm.2011.06.063

Chitsaz, M., Cordeau, J. F., & Jans, R. (2019). A Unified Decomposition Matheuristic for Assembly, Production, and Inventory Routing. *INFORMS Journal on Computing, 31*(1), 134–152. https://doi.org/10.1287/ijoc.2018.0817

Duc, D. N., & Nananukul, N. (2020). A Hybrid Methodology Based on Machine Learning for a Supply Chain Optimization Problem. *Journal of Physics: Conference Series, 1624*, 052022. https://doi.org/10.1088/1742-6596/1624/5/052022

Federgruen, A., & Tzur, M. (1999). Time-partitioning heuristics: Application to one warehouse, multi-item, multiretailer lot-sizing problems. *Naval Research Logistics, 46*(5), 463–486. https://doi.org/10.1002/(SICI)1520-6750(199908)46:5<463::AID-NAV2>3.0.CO;2-S

Gen, M., & Syarif, A. (2005). Hybrid genetic algorithm for multi-time period production/distribution planning. *Computers & Industrial Engineering, 48*(4), 799–809. https://doi.org/10.1016/j.cie.2004.12.012

Geoffrion, A. M., & Powers, R. F. (1995). Twenty Years of Strategic Distribution System Design: An Evolutionary Perspective. *Interfaces, 25*(5), 105–127. https://doi.org/10.1287/inte.25.5.105

Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research, 11*(1), 49–66. https://doi.org/10.1016/0305-0548(84)90007-8

Gong, W., & Fu, Z. (2010, December). ABC-ACO for perishable food vehicle routing problem with time windows. In *2010 international conference on computational and information sciences* (pp. 1261-1264). IEEE. https://doi.org/10.1109/ICCIS.2010.311

Hemmati, M., & Smith, J. C. (2016). A mixed-integer bilevel programming approach for a competitive prioritized set covering problem. *Discrete Optimization, 20*, 105–134. https://doi.org/10.1016/j.disopt.2016.04.001

Iassinovskaia, G., Limbourg, S., & Riane, F. (2017). The inventory-routing problem of returnable transport items with time windows and simultaneous pickup and delivery in closed-loop supply chains. *International Journal of Production Economics, 183*, 570–582. https://doi.org/10.1016/j.ijpe.2016.06.024
Infante, D., Paletta, G., & Vocaturo, F. (2009). A ship–truck intermodal transportation problem. *Maritime Economics & Logistics, 11*(3), 247–259. https://doi.org/10.1057/mel.2009.6

Jie, L., Sava, A., & Xie, X. (2005). Performance Evaluation and Optimization of a Two-Stage Production-Distribution System with Batch Orders and Finite Transportation Time. *IFAC Proceedings Volumes, 38*(1), 331–336. https://doi.org/10.3182/20050703-6-cz-1902.01477

Lee, C. Y., ÇEtinkaya, S., & Jaruphongsa, W. (2003). A Dynamic Model for Inventory Lot Sizing and Outbound Shipment Scheduling at a Third-Party Warehouse. *Operations Research, 51*(5), 735–747. https://doi.org/10.1287/opre.51.5.735.16752

Melo, R. A., & Wolsey, L. A. (2012). MIP formulations and heuristics for two-level production-transportation problems. *Computers & Operations Research, 39*(11), 2776–2786. https://doi.org/10.1016/j.cor.2012.02.011

Min, H., & Zhou, G. (2002). Supply chain modeling: past, present and future. *Computers & Industrial Engineering, 43*(1–2), 231–249. https://doi.org/10.1016/s0360-8352(02)00066-9

Miranda, P. L., Morabito, R., & Ferreira, D. (2019). Mixed integer formulations for a coupled lot-scheduling and vehicle routing problem in furniture settings. *INFOR: Information Systems and Operational Research, 57*(4), 563–596. https://doi.org/10.1080/03155986.2019.1575686

Patiño Chirva, J. A., Daza Cruz, Y. X., & Lopez-Santana, E. R. (2016). A Hybrid Mixed-Integer Optimization and Clustering Approach to Selective Collection Services Problem of Domestic SolidWaste. *Ingeniería, 21*(2), 235-257. https://doi.org/10.14483/udistrital.jour.reveng.2016.2.a09

Rafie-Majd, Z., Pasandideh, S. H. R., & Naderi, B. (2018). Modelling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm. *Computers & Chemical Engineering, 109*, 9–22. https://doi.org/10.1016/j.compchemeng.2017.10.013

Rizk, N., Martel, A., & Ramudhin, A. (2006). A Lagrangean relaxation algorithm for multi-item lot-sizing problems with joint piecewise linear resource costs. *International Journal of Production Economics, 102*(2), 344–357. https://doi.org/10.1016/j.ijpe.2005.02.015

Sun, L., Rangarajan, A., Karwan, M. H., & Pinto, J. M. (2015). Transportation cost allocation on a fixed route. *Computers & Industrial Engineering, 83*, 61–73. https://doi.org/10.1016/j.cie.2015.02.004

Swenseth, S. R., & Godfrey, M. R. (2002). Incorporating transportation costs into inventory replenishment decisions. *International Journal of Production Economics, 77*(2), 113–130. https://doi.org/10.1016/S0925-5273(01)00230-4

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. https://doi.org/10.1016/S0260-8774(00)00187-4

Thomas, D. J., & Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research, 94*(1), 1–15. https://doi.org/10.1016/0377-2217(96)00098-7
Wang, Y., Ma, X., Xu, M., Wang, L., Wang, Y., & Liu, Y. (2015). A Methodology to Exploit Profit Allocation in Logistics Joint Distribution Network Optimization. *Mathematical Problems in Engineering*, 2015, 1–15. [https://doi.org/10.1155/2015/827021](https://doi.org/10.1155/2015/827021)

Xu, Y., & Jiang, W. (2014). An Improved Variable Neighborhood Search Algorithm for Multi Depot Heterogeneous Vehicle Routing Problem based on Hybrid Operators. *International Journal of Control and Automation*, 7(3), 299–316. [https://doi.org/10.14257/ijca.2014.7.3.29](https://doi.org/10.14257/ijca.2014.7.3.29)

Received: 2 September, 2021; Revised: 11 October, 2021; Accepted: 11 November, 2021.

Rad je primljen: 02.09.2021. Korigovan: 11.10.2021. Prihvaćen: 11.11.2021.