HYBRID HEURISTICS FOR INFINITE PERIOD INVENTORY ROUTING PROBLEM

Ronald G. Askin

Mingjun Xia
School of Computing, Informatics & Decision Systems Engineering
Arizona State University
699 S Mill Ave, Tempe, AZ, 85287-5906
E-mail: Ron.Askin@asu.edu & Mingjun.Xia@asu.edu

Abstract

In this paper, we address a one-to-many distribution network inventory routing problem over an infinite planning horizon. Each retailer has an independent, random demand, and the distribution center uses capacitated vehicles for routing delivery. The demand at each retailer is relatively small compared to the vehicle capacity. A novel mathematical model is given to simultaneously decide the optimal routing tours to retailers and routing frequencies of each route. Several heuristics are developed to solve large scale instances of the problem.

1 Introduction

The need for the integration and coordination of various components in a supply chain has been recognized as an important factor in many companies to remain competitive. Inventory management and transportation are two of the key logistical drivers of Supply Chain Management (SCM). The coordination of these two drivers is known as the Inventory Routing Problem (IRP). IRP can be interpreted as an enrichment of Vehicle Routing Problem (VRP) to include inventory concerns.

Numerous research studies have addressed in IRP and many heuristics have been developed to solve practical problems. However, due to the limitation of space, we will only mention the most relevant or representative papers here. Interested readers can find more details in the following three review papers.

Ekasioglu et al. [10] presented a methodology for classifying the literature of VRP. Eight concluding remarks are addressed at the end of the paper, especially about the research opportunities and trends in VRP. Andersson et al. [2] described industrial aspects of IRP in maritime and road-based transportation. A classification and comprehensive literature review of current state of the research, and trends within both industry and research were presented. As noted by both papers, VRP/IRP is important, however, no clear definition exists and there is a gap between research and academia. Baita et al. [4] reviewed the available literature on a class of problems denoted as Dynamic Routing And Inventory (DRAI) problems. Three characteristics are simultaneously considered by this class of papers: Dynamicity, Routing and Inventory.

Bard et al. [7] presented a comprehensive decomposition scheme for IRP. Viswanathan and Mathur [21] considered a multiple-product IRP model and derived a stationary nested joint replenishment policy heuristic. Moin et al. [18] addressed IRP in a many-to-one multi-product multi-period distribution network and used a hybrid genetic algorithm to solve it. Zachariadis et al. [23] proposed an integrated local search method for IRP by insertion and removal of replenishment point, and then used Tabu search for further improvement. Yu et al. [22] solved an IRP based on an approximate model of the problem and Lagrangian relaxation. However, all those papers mentioned in this paragraph assume either constant demand rate or known deterministic demand over a finite time period at each retailer.

A policy consisting of a partition of the retailers into disjoint and collectively exhaustive sets where each set is served on a separate route is called Fixed Partition Policy (FPP). Two important reasons make FPP of interest: simple structure and cost-effectiveness. Chan et al. [8] analyzed the asymptotic effectiveness of FPP and those employing zero inventory ordering. They provided worst case as well as probabilistic bounds under a variety of probabilistic assumptions. Anily and Bramel [3] had a probabilistic analysis of the similar IRP and demonstrated an asymptotically 98.5% effective lower bound. Zhao et al. [24] provided a partition approach to IRP and derived a lower bound of the long-run average cost of any feasible strategy. Zhao et al. [25] used joint FPP and Power-Of-Two (POT) to solve an IRP in a three-echelon logistics system. They also developed a Variable Neighborhood Search (VNS) algorithm to find the retailer’s optimal partition. Li et al. [15] compared the performance of direct shipping strategy and Fixed Partition Policy (FPP), their analysis is helpful to decide which shipping strategy to use based on system parameters.

Greedy Randomized Adaptive Search Procedure (GRASP) is a recently exploited method combining the power of greedy heuristics, randomization, and local search. It is a multi-start two-phase meta-heuristic for combinatorial optimization proposed by Feo and Resende [11], basically consisting of a construction phase and a local search improvement phase. A path-relinking phase can be added/embedded to improve the basic GRASP. Villegas et al. [20] considered a VRP using GRASP, they also use path-relinking as a post-optimization procedure.

Besides the traditional IRP, some researchers add more considerations in their models. For example: IRP with backlogging [1], IRP with production consideration [6], inclusion of the pricing decisions in IRP [17], and allowing lateral transfers of vehicles and inventory [19].

From our literature review, what is missing in IRP research includes considering uncertain demands at retailers, variable routing frequency, and consideration of non-linear characteristics of routing cost and lead time. Most previous research focuses on a single or finite time period with deterministic demand, number of routes and routing frequencies are generally fixed as parameters in the decision model.

The major contribution of our research is to propose a novel model to consider a routing problem with random demand at retailers. In the routing stage, we consider truck capacity and distance limitation, and inventory costs. This model will simultaneously decide the optimal routing tours to retailers and routing frequencies of each route. IRP is NP-hard in the strong sense [4], exact methods usually deal with the capacitated problem with no distance constraints and no empty routes allowed. In this paper, we derive four efficient hybrid heuristics for solving large scale IRP and compare the performances between these heuristics. In using a genetic algorithm, we propose some modified chromosome representations to overcome limitations we incurred with alternate specifications.

2 Problem description and mathematical formulation

We consider a single-product distribution network. The network consists of one distribution center (*DC*) and multiple retailers (*R*). Each retailer has an independent demand for the product. Multiple-product network could be considered provided a common demand measure exists and common deliveries are used, safety stock expressions are updated accordingly.

The goal of our decision problem is to decide routing tours to each retailer and routing frequencies of each tour so that the total routing and inventory cost is minimized over an infinite planning horizon. The *DC* owns multiple homogenous capacitated vehicles, each routing tour should start and end at the *DC*. While demand is random we seek to form standard tours and frequencies. Individual orders will vary based on recent usage and vehicle capacity will be considered to ensure a high probability of being able to meet demand on each route trip. We assume routing frequencies fall in a discrete set such as daily, every other day, weekly and biweekly.

In our problem, the total cost is a summation of routing cost over each trip and inventory cost at each retailer. Routing cost of one trip contains a predetermined fixed cost and a variable cost depending on total distance of this trip. Inventory at each retailer contains both cycle inventory and safety stock. Lead time is assumed to be a function of routing frequency and distance. Detail formulation will be provided later in this section.

2.1 Notation and Decision Variables

Indices

R : set of retailers, $R_0 = R \cup \{DC\}$

V : set of tours

N : set of available routing frequencies

Parameters

C : default vehicle's capacity

D : distance limit for each route
p : speed of the default vehicle
a : fixed cost of using one vehicle at DC
c : unit cost per mile
M : a big positive number
z_α : left α -percentile of standard normal random variable Z.
d_{ij} : distance between node i and node j, subscript "0" represents the DC and other positive integers represent retailers
f_n : routing frequency at level n
h_r : holding cost yearly per unit at retailer r
μ_r : mean of yearly total demand at retailer r
σ^2_r: variance of yearly total demand at retailer r

Variables

x_{ijv} : 1 if i immediately precedes j in route v, 0 otherwise
y_{rv} : 1 if use route v to supply demand at retailer r, 0 otherwise
z_{vn} : 1 if route v has routing frequency at level n, 0 other wise

2.2 Problem formulation

Minimize

$$\sum_{v\in V}\left(a+c\sum_{i,j\in R_0} d_{ij}x_{ijv}\right)\left(\sum_{n\in N} f_n z_{vn}\right)+ \sum_{r\in R} h_r\left[\sum_{v\in V} y_{rv}\left(\frac{0.5\mu_r}{\sum_{n\in N} f_n z_{vn}}+z_\alpha\sigma_r\sqrt{\frac{1}{\sum_{n\in N} f_n z_{vn}}+\frac{\sum_{i,j\in R_0} d_{ij}x_{ijv}}{p}}\right)\right] \tag{1}$$

Subject to:

$$\sum_{j\in R_0} x_{ijv}=\sum_{j\in R_0} x_{jiv} \qquad \forall i\in R_0, v\in V \tag{2}$$

$$\sum_{i,j\in B} x_{ijv}\le |B|-1 \qquad \forall B\subseteq R, |B|\ge 2, v\in V \tag{3}$$

$$\sum_{j\in R_0} x_{jrv}=y_{rv} \qquad \forall r\in R, v\in V \tag{4}$$

$$\sum_{v\in V} y_{rv}=1 \qquad \forall r\in R \tag{5}$$

$$\sum_{n\in N} z_{vn}=1 \qquad \forall v\in V \tag{6}$$

$$\sum_{r\in R}\left(\frac{\mu_r y_{rv}}{\sum_{n\in N} z_{vn} f_n}\right)\le C \qquad \forall v\in V \tag{7}$$

$$\sum_{i,j\in R_0} d_{ij}x_{ijv}\le D \qquad \forall v\in V \tag{8}$$

$$x_{ijv}, y_{rv}, z_{vn}=\{0,1\} \qquad \forall i,j\in R_0, v\in V, r\in R, n\in N \tag{9}$$

The objective function has two components: routing cost and inventory cost. Inventory cost includes both cycle inventory to meet foreseeable demand and safety stock to overcome uncertain demand. Constraint 2 is network conservation, for each arc entering one node, it should also leave this node. Constraint 3 eliminates subtours. Constraints 4-6 guarantee each retailer is assigned to one and only one route. Constraints 7 and 8 are vehicle capacity and route distance limits. Constraints 9 are discrete/binary variable constraints.

To simplify the above formulations, let:

$\gamma_v = \sum_{n \in N} f_n z_{vn}$: number of trips for route v in one year.

$d_v = \sum_{i,j \in R_0} d_{ij} x_{ijv}$: distance of route v for $\forall v \in V$.

$Q_{rv} = \dfrac{\mu_r y_{rv}}{\gamma_v}$: average delivery size for retailer r through route v each time.

$lt_r = \dfrac{1}{\sum_{v \in V} \gamma_v y_{rv}} + \dfrac{\sum_{v \in V} d_v y_{rv}}{p}$: lead time for the retailer r. Lead time is a function of routing route frequency (first component) and route distance (second component).

Then the objective function is to minimize:

$$IRC = \sum_{v \in V} (a + cd_v)\gamma_v + \sum_{r \in R} h_r \left(0.5 \sum_{v \in V} Q_{rv} + z_\alpha \sigma_r \sqrt{lt_r} \right) \tag{10}$$

Subject to constraints (2) – (6), (9), and:

$$\sum_{r \in R} Q_{rv} \le C \qquad \forall v \in V \tag{11}$$

$$d_v \le D \qquad \forall v \in V \tag{12}$$

3 Problem Characteristics

3.1 Optimal Delivery Frequency

If one routing tour is decided, then retailers in this route and the routing cost per trip in this route are known. To decide the optimal routing frequency is to Minimize:

$$(a + cd_v)\gamma_v + \sum_{r \in S_v} h_r \left(\frac{0.5\mu_r}{\gamma_v} + z_\alpha \sigma_r \sqrt{\frac{1}{\gamma_v} + \frac{d_v}{p}} \right) \qquad \forall v \in V \tag{13}$$

subject to the vehicle capacity constraints, where S_v is the set of retailers serviced by this route v. As a discrete variable having relatively few available values, we can simply try each value and use the one minimizing the total cost. The nonlinearity of the objective makes it difficult to obtain a closed form optimality expression, but the first and second order derivatives are provided in the **Appendix** for nonlinear search techniques.

3.2 Upper/Lower Bounds for the Number of Tours

In this research, we do not know the optimal number of tours needed. If using full-truck-load to delivery products as often as we can (at the maximum allowed frequency, $\gamma_{\max}$), a lower bound is generated:

$$V_{LB} = \frac{\sum_r \mu_r}{\gamma_{\max} C}$$

And if using one individual route for each retailer, then an upper bound is generated: $V_{UB} = N$. This upper bound is used later on in genetic algorithm to create chromosomes.

3.3 Upper/Lower Bounds for the Objective Values

The major benefit of routing comes from reduction in delivery cost. If there are no distance/capacity limitations, nearest neighbors will be merged into one tour. In an ideal case, delivery distance to one retailer is 1 + 1/ (N+1) times the distance between nearest neighbors, where N is the total number of retailers. Smallest total number of trips required is total demand over all retailers dividing by a truck capacity. Let IRC be total inventory routing cost, so a lower bound for the objective value is generated as:

$$IRC_{LB} = a\left(\frac{\sum_{r\in R} D_r}{C}\right) + \sum_{r\in R} c\left(1+\frac{1}{N+1}\right) d_r \gamma_r + \sum_{r\in R} h_r \left(\frac{D_r}{2\gamma_r} + z_\alpha \sigma_r \sqrt{\frac{1}{\gamma_r} + \frac{\left(1+\frac{1}{N+1}\right) d_r}{p}}\right) \quad (14)$$

where d_r is distance to its nearest neighbor for each retailer. The value of second part can be found using methods introduced in previous section.

The above lower bound is very tight, so an estimation of total cost is also generated by considering delivery distance to each retailer as D / n, where D is the distance limit and n is the average number of retailers in one route. This estimation formula is not a lower bound, and is only used to estimate the possible optimal total routing cost.

$$IRC_e = \sum_{r\in R}\left[\left(\frac{a+cD}{n}\right)\gamma_r + h_r\left(\frac{D_r}{2\gamma_r} + z_\alpha \sigma_r \sqrt{\frac{1}{\gamma_r} + \frac{D}{np}}\right)\right] \quad (15)$$

Any feasible solution is an upper bound, a simple solution is using all direct-shipping. In this case, each retailer has one individual route, and the frequency is selected to minimize this individual tour.

4 Heuristics

This proposed IRP belongs to the class of NP-hard problems. In this section, several heuristics are developed to solve this IRP problem for medium and large instances. The basic idea is to generate fixed partitions of retailers and use one vehicle to serve one group of retailers. After a routing tour is determined within a group of retailers, routing frequencies are selected from available frequency set.

4.1 Modified Sweep Method (MS)

Evidence indicates that the sweep method for routing vehicles is computationally efficient and produces an average gap from optimality of about 10 percent [5]. This gap may be acceptable where results must be obtained in short order and good solutions are needed as opposed to optimal ones.

We modify the simple sweep method by considering specific characteristics in this problem: optimize routing tour after inserting one new retailer, optimize routing frequency within one route, start from each retailer and sweep both clockwise and anti-clockwise. The procedures of our modified sweep method are as follows:

Procedures:

1. Locate the *DC* and all retailers on a map or grid.
2. Extend a straight line from the *DC* in any direction. Rotate the line clockwise until it intersects one retailer. For the first retailer the line intersects, build one individual route for this retailer (retailer 1 in Figure 1).

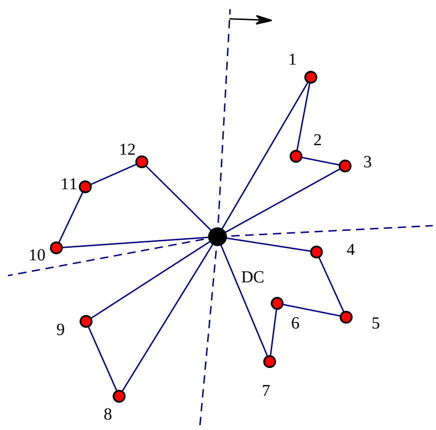

Figure 1: Example sweep result

3. Continue to rotate the line until next retailer is reached, insert new retailer in current route using nearest insertion method and try to improve the new route by 2-opt method. After including new retailer and deciding new optimal route in current route, check all constrains and recalculate the optimal routing frequency and total cost.
4. If adding the new retailer to current route can reduce total cost and all constraints are met, add this retailer current route; otherwise, create a new route for the new retailer.
5. Continue until all retailers are assigned.

In step 4, after checking all constraints if adding the next retailer, two cases are compared to finally decide whether to add this new retailer or not: one is to add this new retailer resulting in one longer route, the other is the previous route and a new individual route for this new retailer. Let:

RC : routing cost per trip
IC : inventory cost
IRC : total inventory routing cost
γ : optimal routing frequency
v : previous route
i : a new retailer
$v+i$: a longer route after adding retailer i
Case1: $IRC_{v+i} = RC_{v+i}\gamma_{v+i} + IC_{v+i}$
Case2: $IRC_v + IRC_i = RC_v\gamma_v + RC_i\gamma_i + IC_v + IC_i$

Whether to add retailer i to the previous route depends on the value of these two cases, the one with smaller value is the solution.

The procedure stated above starts rotation at a random retailer location, one issue may arise: suppose we rotate the line clockwise in the above figure, then the left retailer (retailer 12) will be in a different route from retailer 1 almost for sure, but it may be better to group these two retailers. In order to solve this issue, we will do the sweep algorithm $2N$ times, where N is the number of retailers. Use one different retailer as a starting rotation points, sweep both clockwise and counterclockwise for each starting point, and then choose the best solution among these $2N$ solutions as our final solution.

Pseudo code:

```
For each retailer i: select it as the starting point, extend a straight line from the DC to
retailer i, do:
  Rotate this line clockwise (and counterclockwise) until it inserts one retailer j, do:
      If adding retailer j to current route violate truck capacity constraint:
          Start a new route;
      Else
          Insert retailer j to current route, and use 2-opt to improve routing tour;
              If the improved tour distance violate the routing distance constraint:
```

Start a new route
Else
If using separate routes, the total cost is less:
Start a new route;
Else
Insert retailer j to current route;
End If
End If
End If
Record the total cost if using retailer i as starting point.
Compare and select the smallest cost and use it as the final solution.

4.2 Tabu Search – Simulated Annealing Method (TS-SA)

Tabu search (TS) [12] and Simulated Annealing (SA) [9] are two successful meta-heuristic solution approaches to solve hard combinatorial problems. The most important feature of Tabu search is its ability to avoid search cycling by systematically preventing moves that generate the solutions previously visited in the solution space. Simulated annealing allows the search to proceed to a neighboring state even if the move causes the value of the objective function to become worse, and this allows it to prevent falling in local optimum traps. TS-SA approach will join these two advantages [14]. Javid and Azad [14], Lin et al. [16] used this TS-SA in their research for VRP and Location problem respectively. Our TS-SA heuristic is similar to theirs but differs in neighborhood constructions and considering frequency selection.

T_0 : Initial temperature
T : Current temperature
α : Decreasing rate of current temperature (cooling schedule), $0 < \alpha < 1$
FT: Freezing temperature (temperature at which the desired energy level is reached)
MaxNum : Maximum number of accepted solutions at each temperature
Num : Counter for number of accepted solutions at each temperature
NOIMPROVE : Maximum number of iterations to run algorithm
Noimprove : Current number of iterations that the best solution is not improved
X_0 : Initial solution
X : Current solution in algorithm
X_{nh} : Solution which is selected in neighborhood of X in each iteration
X_{best} : Best solution obtained in algorithm
$C(X)$: Objective function value for solution X

Procedures:

1. Take the initial solution X_0, and set $X_{best} = X_0$, $X = X_0$, $T = T_0$. $Num = Noimprove = 0$. The initial solution can be generated randomly or just use the output of modified sweep method.

2. Generate a feasible solution X_{nh} in the neighborhood of X using moves.
3. If the move generated X_{nh} in the tabu list and X_{nh} is not the best solution found so far, go back to generate another neighborhood, and update the tabu list.
4. $Num = Num+1$.Update the tabu list and let $\Delta C = C(X_{nh}) - C(X)$.
   If $\Delta C \le 0$, then $X = X_{nh}$. If $C(X_{nh}) < C(X_{best})$, $X_{best} = X_{nh}$, $Noimprove = 0$.
   If $\Delta C > 0, y \leftarrow U(0,1), z = T / \Delta C$. If $y < z$ and $\Delta C < T$, then $X = X_{nh}$, In this case, we may move to a solution that is worse than current solution.
5. Whether $Num < MaxNum$? If yes, go to Step 2; if not go to Step 6.
6. $Noimprove = Noimprove+1$, $T = \alpha T$.
7. Is the stopping criterion ($T < FT$ *or* $Noimprove < NOIMPROVE$) matched? If yes, stop; or else go to Step 2.

Neighborhoods of current solution are generated using the following moves. Before stating these moves, we have two definitions to declare:

Distance between two routes: For all pairs of retailers in two different routes, the smallest possible distance between two retailers is called the distance between these two routes. Let S_k be the set of retailers included in route k; D_{ij} be the distance between retailer i and j, and DR_{mn} be the distance between route m and n, then: $DR_{mn} = \arg\min\{D_{ij}\} \quad i \in S_m, j \in S_n$

Adjacent route: Two routes are called adjacent if the distance between these two routes is smallest compared to other routes (or within some predetermined value).

Move 1: Select two retailers in one route and then exchange their delivery order.
Move 2: Select two retailers from two adjacent routes and then exchange them.
Move 3: Select one retailer randomly and insert it to an adjacent route.
Move 4: Select one retailer randomly and then open a new individual route for it.

Pseudo code:

```
Take the initial solution X0, and set Xbest = X0, X = X0, T = T0, Noimprove = 0.
While (T > FT & Noimprove < NOIMPROVE ), do
{
  For (Num = 0; Num <= MaxNum; Num++), do:
  {
      Generate a neighborhood solution Xnh.
      If (Xnh is not in the tabu list)
      {
          Update the tabu list; ΔC = C(Xnh) – C(X).
          If (ΔC ≤ 0)
          {
              X = Xnh.
              If ( C(Xnh) < C(Xbest) )
```

$X_{best} = X_{nh}$, $Noimprove = 0$.
}
Else
{
$y \leftarrow U(0,1), z = T / \Delta C$.
If ($y < z$ and $\Delta C < T$)
$X = X_{nh}$
}
}
Else If ($X_{nh} = X_{best}$)
$X = X_{nh}$
}
$Noimprove = Noimprove+1$.
$T = \alpha T$.
}

4.3 Integrated Local Search Method (ILS)

A distinction of this research is to simultaneously consider routing tour and routing frequency over an infinite planning horizon while traditional routing solution methods usually only focus on routing tour. In order to capture routing frequency, we propose this integrated local search method.

The basic idea is to generate an initial solution where each retailer is serviced by one individual tour, and then try to merge retailers into one route. The optimal routing frequency for each retailer under an individual tour is called the **natural frequency** for this retailer. This heuristic is also suitable if **natural frequency** is given in reality, for example, some retailers receive orders daily/weekly.

When calculating the natural frequency, routing cost per trip is calculated as one fixed vehicle cost plus variable cost from *DC* to the retailer. This is considering the performance of one retailer in a joint routing tour with multiple-retailers. The routing cost for one retailer in such a tour is only part of one fixed cost and some insertion distance from previous/next neighborhood. In the computing step, we also introduce another two scenarios: “Fixed cost + Variable cost (twice the distance from the *DC*)” and "[Fixed cost + Variable cost (Distance limitation)] / Average number of retailer in one route", all three scenarios' results are compared. Scenario two implicitly assumes single retailer rates.

Since the available values for routing frequency are discrete (daily, once other day, weekly, biweekly and assume 1 year = 350 days), the natural frequency for each retailer will be found by searching for the lowest cost policy over these options. Whether to merge two retailers depends on two factors: the distance between these two retailers and similarity in natural frequency. If two close retailers have similar natural frequency, using one vehicle to serve both of them will reduce the total cost.

Procedures:

1. Calculate natural frequency for each retailer.
2. Divide all retailers into different groups based on natural frequencies, retailers in the same group will have the same natural frequency. In this research, four groups will be generated with routing frequency to be 350, 175, 50, and 25, respectively. Call these four groups to be G_1, G_2, G_3 and G_4.
3. Use embedded modified sweep method to merge retailers in group G_1 (the group with largest routing frequency).
4. After generating tours for all retailers in group G_1, try to insert other retailers in other groups (in the order of G_2, G_3 and G_4) in current routes. The motivation to do this step is because of the possibility of the following case:

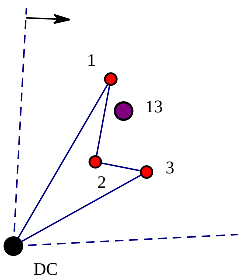

Figure 2: Insertion example

   In this case, one route is generated to serve retailer 1, 2, 3 and all these three retailers have the same routing frequency. The distance limitation is validated if we want to add any other retailer from group G_1. However, one retailer (retailer 13) is very close to retailer 1 and has a natural frequency smaller than G_1. If adding retailer 13 does not violate any distance/capacity constraint, the total cost may be less if inserting retailer 13 into current route. This is also the reason why we start from the largest frequency group G_1. Merging a retailer with smaller natural frequency to a route with larger routing frequency will reduce the average cycle inventory level at this retailer, so the inventory cost will be reduced. And since we use extra truck capacity and little additional variable routing cost to serve another retailer, the total routing cost will also be reduced.
5. Repeat the same process of step 3 and 4 for retailers in group G_2, G_3 and G_4 respectively.
6. *After generating an initial solution from the above five steps, we can add an improvement step using Tabu search. Neighborhoods can be generated by two moves introduced in section 4.2, however, negative gain is not allowed here. A solution is updated only if one neighborhood has smaller objective value. We will compare the results whether adding step 6 or not the computing step.

In the Modified Sweep method, the final solution should be several disjoint routes since we add retailers by sweeping one line clockwise. However, in this method, the final solution may have a structure as shown in Figure 3. Two routes are overlapped but with different routing frequency. For example, route 1 deliveries products daily, while route 2 deliveries products weekly.

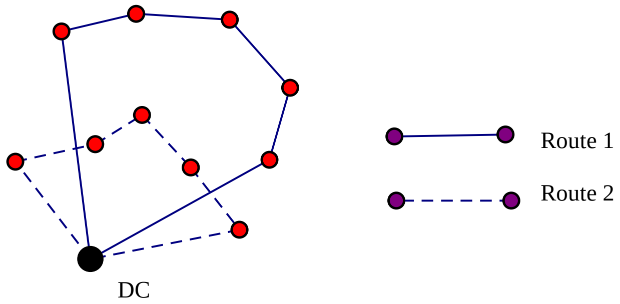

Figure 3: Routing structure

Pseudo code:

```
Calculate natural frequency for each retailer.
Divide all retailers into different groups based on their natural frequencies.
Let S = number of groups (S = 4).
For (i = 1; i <= S; i++)
{
   If (Group i is not empty)
   {
       Use modified sweep method to merge retailers in Group i.
       Same procedure as in modified sweep, sweep clockwise and anti-clockwise from
       each retailer.
       Find the optimal starting point r and direction.
       For(j = i+ 1; j <= S;  j++ )
       {
           If (Group j is not empty)
               Try to insert retailers in Group j to existed routes;
       }
   }
}
(Improve current solution by Tabu search by setting it as the initial solution.)
```

4.4 Hybrid Genetic Algorithm Method (HGA)

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. A simple procedure for traditional GA is shown as follows:

Procedures:

1. Set $t = 0$. Initialize Population $P(t)$ with randomly constructed solutions. Alternatively use results from heuristics (i.e., modified sweep method) as partial population.
2. Evaluate the feasibility and fitness function of individuals included in $P(t)$.
3. Apply Crossover and Mutation operators to obtain a set $C(t)$ of candidates that can satisfy problem constraints.
4. Evaluate the set $C(t)$ of candidate and select the best individuals with respect to fitness value to add to new Population $P(t+1)$. The new population consists of the best *PS* (population size) chromosomes from $P(t)$ and $C(t)$.
5. $t = t + 1$, while stopping criteria are not met do, go back to step 2.
6. End and keep the best individual of the last population as the solution of the problem.

The idea for our hybrid heuristic is to use a genetic algorithm (GA) to generate/update a fixed partition for all retailers. A TSP is solved within each partition and optimal delivery frequency is selected accordingly. In a fixed partition policy (FPP), the retailers are partitioned into disjoint and collectively exhaustive sets. Each set of retailers is served independently of the others and at its optimal replenishment rate. The framework is shown in Figure 4, where GA is used to generate and update fixed partition, TSP is solved by 2-opt heuristic.

4.4.1 Chromosome Representation

In our research, the real number of vehicles used is an unknown variable, but the maximum number will be the number of retailers, in this case, each retailer is serviced by one individual route. The length of a chromosome is equal to the number of retailers N. Each gene of the chromosome is related to a retailer and is assigned to an integer number between 1 and N. If the ith gene is assigned to integer m, for instance, then it means that retailer i is served by vehicle m.

1	2	3	4	5	6	7	8	9	10
2	1	4	2	1	1	4	3	3	2

The above chromosome represents a 4-vehicle solution, vehicle 1 services retailer 2, 5 and 6, vehicle 2 services retailer 1, 4 and 10, etc.

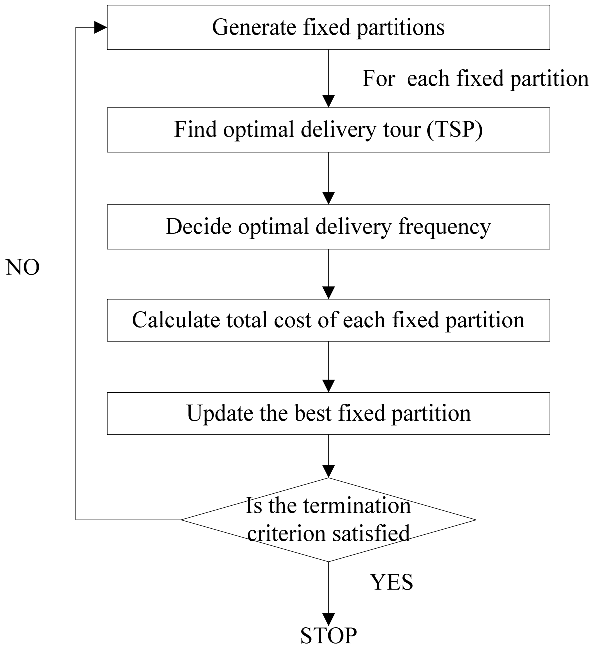

Figure 4: HGA framework

4.4.2 Chromosome Justification

The chromosome representation introduced above is easy to understand, but there will be an issue in practice. For example, the following two chromosomes actually represent the same solution. It's a 4-vehicle solution with one vehicle services retailer 1,4,10; one vehicle services retailer 2,5,6, one vehicle services retailer 8,9 and one vehicle services retailer 3,7.

1	2	3	4	5	6	7	8	9	10
2	1	4	2	1	1	4	3	3	2
5	1	3	5	1	1	3	2	2	5

The differences in chromosomes' representations come from the order of different routes. To deal with this symmetry and make the further calculation easier, we will do a chromosome justification every time after generating a new chromosome.

Justification: Number retailers from 1 to *N*. Following the order of retailers, each retailer is assigned to the smallest available vehicle number.

By adopting this justification, the above two chromosomes will be modified to:

1	2	3	4	5	6	7	8	9	10
1	2	3	1	2	2	3	4	4	1

4.4.3 Crossover

Crossover is a mechanism in which the information between two chromosomes is exchanged randomly. We can use two-point crossover operator, for example:

1	2	3	4	5	6	7	8	9	10
1	2	3	**1**	**2**	**2**	3	4	4	1
1	2	3	**4**	**2**	**1**	5	4	5	3

After crossover:

1	2	3	4	5	6	7	8	9	10
1	2	3	**4**	**2**	**1**	3	4	4	1
1	2	3	**1**	**2**	**2**	5	4	5	3

Or, we can use one-point crossover operator, for example:

1	2	3	4	5	6	7	8	9	10
1	2	3	1	**2**	**2**	**3**	**4**	**4**	**1**
1	**2**	**3**	**4**	2	1	5	4	5	3

After crossover:

1	2	3	4	5	6	7	8	9	10
1	2	3	1	2	1	5	4	5	3
1	**2**	**3**	**4**	**2**	**2**	**3**	**4**	**4**	**1**

4.4.4 Mutation

In a mutation operator, each gene can change to a different integer number with a defined probability, two examples:

1	2	3	4	5	6	7	8	9	10
1	2	3	1	2	2	3	4	4	1

Example 1

1	2	3	4	5	6	7	8	9	10
1	2	3	1	**4**	2	3	4	4	1

Example 2

1	2	3	4	5	6	7	8	9	10
1	2	3	1	**5**	2	3	4	4	1

In the first example, the number of routes does not change, but the real routes change. In the second example, by assigning retailer 5 to route 5, the original 4-route solution becomes a 5-route solution. Also we check if justification is necessary whenever a new chromosome is generated.

4.4.5 Fitness Function (*ff*)

The fitness score is a possibly-transformed rating used by the genetic algorithm to determine the fitness of individuals for mating. In this heuristic, we use objective function (total cost) directly as fitness score. It is probabilistically possible for infeasible solutions to survive. A penalty value (a big positive number *M*) is applied to the fitness function without removing infeasible solutions. The logic behind this is that an optimal solution may exit with high probability near an infeasible solution. However, at the same time, the proposed algorithm will record the best feasible solution. Since the fitness function here is actually the total cost, then the smaller the value is, the better the chromosome is.

4.4.6 Selection

The roulette wheel selection operation [13] is adopted to choose some chromosomes to undergo genetic operations. The approach is based on an observation that a roulette wheel has a section allocated for each chromosome in the population, and the size of each section is proportional to the chromosome's fitness: the fitter the chromosome, the higher the probability of being selected. Although one chromosome has the highest fitness, there is no guarantee it will be selected. But on average, a chromosome will be chosen with the probability proportional to its fitness. Suppose the population size is *PS*, then the selection procedure is as follows:

1. Calculate the total fitness of the population as *FF*.
2. Calculate the selection probability sp_i for each chromosome X_i: $sp_i = \frac{FF - ff(X_i)}{FF(PS-1)}$
3. Calculate the cumulative probability qp_i for each chromosome X_i: $qp_i = \sum_{j=1}^{i} sp_j$
4. Generate a random number r from a uniform distribution in the range (0, 1].
5. If $qp_{i-1} < r \le qp_i$, then chromosome X_i is selected.

4.4.7 TSP

In the first stage, retailers are grouped into several sets and each set is serviced by one vehicle. Within each set, we use 2-opt search method to optimize the delivery tour and a delivery frequency is selected later. For example, by solving TSP, we get the final solution as:

Route 1: {1, 4, 10}		DC-4-10-1-DC
Route 2: {2, 5, 6}	$\Rightarrow$	DC-2-6-5-DC
Route 3: {3, 7}		DC-3-7-DC
Route 4: {8, 9}		DC-8-9-DC

The total cost can be calculated based on the final routing schedule after selecting delivery frequency for each route.

Pseudo code:

```
TG: Number of total generations
SG: Stop if there is no improvement within such generations.
sg = t  = 0;

Initialize the Population P(t)
For (t = 1; t <TG & sg < SG; t++)
{
   Apply Crossover and Mutation operators to obtain a set C(t) of candidates;
   Calculate fitness function value of C(t) and update population to new Population
   P(t+1);
   If (any new generated chromosome performs better than best solution in P(t) )
        sg = 0;
   Else
        sg = ++;

   t ++;
}
```

5 Computational Results

To evaluate the performance of our four heuristics, we use the computational experiments described in this section. Besides, all direct-shipping method is used to calculate upper bound, and total cost estimation and a lower bound are generated using methods introduced in Section 3.

5.1 Parameter Settings

Parameter settings are defined in Tables 1 and 2. Retailers are assumed to be randomly located in a 200-mile by 200-mile square with the distribution center in the center. Number of retailers, holding cost and demand standard deviation were variables shown in Table 3 to form 30 scenarios. Mean retailer demand, service level, vehicle capacity and speed, distance limit (length of daily tour), location of *DC*, and fixed truck cost were held constants. Vehicle capacity is set to be 150, this value is roughly estimated so that one vehicle is used to serve about 10 retailers every two days (Average demand/10 $\approx$ 150). Deliveries may be made daily, every other day, weekly, or biweekly.

Additional parameters were set for the TS-SA and HGA procedures as shown in Table 3. These were selected based on previous study and preliminary experimentation.

Table 1: Problem parameter settings

Name	**Notation**	*Value*	*Remark*
Service level	z_α	1.96	97.50%
Vehicle capacity	C	150	
Distance limit	D	500 miles	
Vehicle speed	p	500 miles/day	
Fixed cost	a	$ 5/truck	
Variable routing cost	cd	c = $ 0.1mile d = distance (miles)	
Available frequency/year	f_n	{25, 50, 175, 350}	1year = 350 days
Location of DC	0	(0, 0)	
Number of retailers	N	{20, 50, 100, 150, 200}	
Locations of retailers	(x, y)	[-100, 100]	Uniform Distribution
Demand mean/year	μ_r	10% Low: [50, 150] 80% Medium: [500, 2000] 10% High: [10000, 25000]	Uniform Distribution
Demand standard variance/year	σ_r	Low: [1, 5] High: [10, 50]	Uniform Distribution
Holding cost	h_r	Low: $ 10/unit year Medium: $ 50/unit year High: $ 100/unit year	

5.2 Results and Analysis

Five random instances were generated for each experiment scenario. All four heuristics were then applied to each instance, and results in the following tables for each scenario are the average of those five random instances. For meta-heuristics, the maximum running time was set to be 3600 seconds (1 hour).

Computational times in seconds are shown in Table 4. Table 5 summarizes the objective value results. The best value for each scenario is shown in bold font.

All heuristics except HGA work well in terms of objective values. The HGA takes the most computational effort and returns the highest average cost. Comparing to all direct-shipping method, using routing to serve sets of retailers will reduce total cost by 25.8% - 51.4%.

Table 2: Experiments scenarios

Scenario (k)	N	h_r	σ_r	Scenario (k)	N	h_r	σ_r
1	20	High	High	16	20	Medium	Low
2	50	High	High	17	50	Medium	Low
3	100	High	High	18	100	Medium	Low
4	150	High	High	19	150	Medium	Low
5	200	High	High	20	200	Medium	Low
6	20	High	Low	21	20	Low	High
7	50	High	Low	22	50	Low	High
8	100	High	Low	23	100	Low	High
9	150	High	Low	24	150	Low	High
10	200	High	Low	25	200	Low	High
11	20	Medium	High	26	20	Low	Low
12	50	Medium	High	27	50	Low	Low
13	100	Medium	High	28	100	Low	Low
14	150	Medium	High	29	150	Low	Low
15	200	Medium	High	30	200	Low	Low

Table 3: Heuristics parameter settings

TS-SA		Hybrid GA	
Name	Value	**Name**	**Value**
T_0	1500	Population size	2N
FT	10	Elite proportion	0.05
α	Uniform: [0.7, 1.0]	Mutation probability	0.05
MaxNum	500	*TG*	3000
NOIMPROVE	5	*SG*	100

Among all heuristics, Modified Sweep method performs the best and HGA method is the worst. Using modified sweep, even the largest case, it only takes 2 minutes and finds a good solution. But HGA takes a long time and generates worse solutions. Two major reasons may explain this result:

1. Even our modified sweep method is quite straightforward, it has some theoretical foundation and captures many important aspect of this routing problem. In our problem, we have distance and capacity constraints, and it is preferred to merge near retailers together for the consideration of shipping, thus we sweep all retailers clockwise and counterclockwise. Every time we decide whether or not to insert a new retailer, we modify the route and use 2-opt to improve the route tour. And we also consider comparing joint tours and separate frequency tours.

Table 4: Computational effect: CPU (sec)

k	MS	TS-SA	ILS1	ILS2	ILS3	ILS+TS	HGA
1	2	48	**0**	**0**	1	44	132
2	14	182	2	**1**	4	298	1216
3	61	247	5	**4**	8	1214	3600
4	146	1958	17	**13**	27	3600	3600
5	267	2063	47	**34**	42	3600	3600
6	2	126	1	**0**	**0**	30	98
7	14	256	**1**	**1**	2	553	1346
8	53	484	6	5	**4**	1663	3600
9	158	496	35	16	**15**	3600	3600
10	286	908	106	49	**35**	3600	3600
11	3	88	**0**	**0**	**0**	36	132
12	14	322	**1**	**1**	**1**	820	822
13	56	537	4	**3**	4	1608	3600
14	116	1273	5	**5**	9	3600	3600
15	237	2753	15	**12**	14	3600	3600
16	2	75	**0**	1	1	44	116
17	13	146	**1**	**1**	**1**	401	1525
18	36	253	2	**1**	8	1660	3600
19	79	470	6	**5**	26	3600	3600
20	171	875	16	**13**	58	3600	3600
21	2	74	**0**	**0**	**0**	59	136
22	9	140	2	2	**1**	165	1212
23	28	314	14	19	**2**	1549	3600
24	57	346	53	61	**20**	3600	3600
25	95	534	163	183	**64**	3600	3600
26	2	127	1	**0**	**0**	32	151
27	7	117	2	2	**1**	1367	1050
28	26	340	14	15	**2**	2553	3600
29	60	364	63	78	**19**	3600	3600
30	97	568	195	207	**77**	3600	3600

ILS1: $a + cd_{or}$
ILS2: $a + 2cd_{or}$
ILS3: $a + cD / n$, where n is the average number of retailers in one route

2. HGA method works by making improvement from operators (crossover and mutation). However, with capacity and distance constraints, there is a high probability that a child from crossover and mutation is infeasible, especially in large instances. If we allow the HGA to run infinitely, it may find the best solution, but this is not efficient.

Table 5: Summary Comparison: Objective values ($1000)

k	MS	TS-SA	ILS1	ILS2	ILS3	ILS+TS	HGA	Lower bound	IRC_e	Direct shipping
1	**45.0**	47.3	58.8	56.6	52.3	47.5	45.3	30.8	41.0	71.9
2	99.5	110.6	128.2	131.9	121.2	**97.2**	109.7	63.4	99.0	174.0
3	**198.2**	215.2	260.2	271.8	233.0	200.9	250.8	123.9	207.0	363.9
4	288.6	**285.7**	388.5	409.2	364.4	290.2	347.6	173.7	307.2	548.0
5	**373.5**	375.1	509.9	525.3	434.7	374.3	485.5	217.4	401.3	723.5
6	**33.4**	34.5	42.0	37.2	47.3	33.9	35.0	20.8	30.5	57.7
7	**84.6**	93.0	107.9	109.5	107.2	86.6	91.0	50.3	81.6	152.8
8	**145.0**	150.2	186.3	200.2	194.9	148.5	175.3	81.8	153.8	279.3
9	**200.8**	215.3	292.3	285.0	267.3	205.3	224.3	114.2	228.9	413.1
10	271.9	272.3	385.8	381.4	358.2	**270.3**	347.2	148.3	307.4	555.0
11	37.5	38.2	43.4	46.4	57.4	38.6	**37.0**	23.4	30.8	52.6
12	68.0	71.4	83.2	88.3	95.1	**68.0**	77.4	41.8	68.8	114.3
13	138.9	138.0	166.2	169.7	175.7	**135.7**	160.3	74.9	140.2	235.5
14	**214.1**	214.5	253.6	265.6	285.1	215.1	264.1	111.4	217.0	369.7
15	274.8	**248.0**	330.5	347.9	347.7	268.9	351.1	139.3	284.4	482.3
16	30.3	31.5	35.1	36.9	41.8	30.4	**29.7**	18.2	23.1	42.4
17	**80.8**	83.6	95.5	99.4	98.1	81.9	87.1	41.9	66.4	119.5
18	142.9	148.3	163.2	167.0	175.6	**133.3**	175.3	65.8	119.5	217.2
19	209.5	209.3	242.8	249.8	260.6	**200.3**	233.1	92.7	180.0	325.8
20	248.4	244.6	302.1	311.7	307.6	**234.3**	333.9	107.7	225.5	408.5
21	**16.2**	16.9	18.3	18.3	23.6	16.4	17.9	8.7	10.9	23.0
22	**36.8**	39.3	42.6	42.6	47.6	37.1	38.8	18.4	26.1	55.4
23	76.3	76.6	90.3	85.9	92.0	**70.1**	90.2	34.5	53.7	112.1
24	108.5	116.9	128.7	125.3	145.0	**102.3**	143.4	47.2	80.0	163.5
25	149.8	156.1	176.9	174.6	194.1	**140.2**	191.6	61.9	109.0	224.5
26	**16.2**	17.1	18.7	17.8	24.4	17.2	16.5	9.0	10.0	21.8
27	**34.2**	36.3	39.9	38.9	50.0	34.6	38.4	16.3	22.6	50.4
28	65.2	68.2	78.8	77.4	81.9	**64.1**	70.3	34.6	56.6	100.0
29	99.4	108.5	118.3	119.0	138.8	**96.1**	136.2	45.5	80.1	153.8
30	139.7	146.9	164.7	163.8	169.0	**134.7**	188.1	50.7	91.6	214.6

LS works very fast in terms of CPU time, but its objective values are much higher than MS. If joint with Tabu search, ILS-TS generates better results than MS in large scenarios, but CPU time increases because of Tabu search step. So we recommend using MS method for IRP in this research stage, and we can also use Tabu search (TS) to further improve results from MS method if necessary.

The saving percentage (1 - best solution / direct-shipping cost) is shown in Table 6. When the holding cost and demand variance decrease, the benefits of routing strategy also decreases. Retailers will prefer to order more products each time when their inventory cost is lower, so the number of retailers in one route will decrease because of capacity limitation. In the extreme case, when the number of retailers in one route is only one, this is equivalent to direct-shipping. Routing strategy will have more benefits if the demand or optimal order size of each retailer is small compared to vehicle capacity.

Table 6: Saving Percentage (%)

h_r	σ_r	$N = 20$	$N = 50$	$N = 100$	$N = 150$	$N = 200$	Average
High	High	37.4	44.2	45.5	47.9	48.4	44.7
High	Low	42.1	44.6	48.1	51.4	51.3	47.5
Medium	High	29.7	40.5	42.4	42.1	48.6	40.7
Medium	Low	29.9	32.3	38.6	38.5	42.6	36.4
Low	High	29.6	33.7	37.5	37.2	37.6	35.1
Low	Low	25.8	32.2	35.9	37.5	37.2	33.7

6 Conclusion

Inventory management and transportation are two major considerations in Supply Chain Management (SCM). The coordination of these two drivers is known as the Inventory Routing Problem (IRP). In this paper, we address a one-to-many distribution network IRP over an infinite planning horizon.

The major contribution of our research is to propose a novel model to consider random demands at retailers and choice of delivery frequency. In the routing stage, we consider both truck capacity and distance limitation, and then simultaneously decide the optimal routing tours to retailers and routing frequencies of each route.

IRP is NP-hard in the strong sense, so several versions of four heuristics are developed to solve medium and large instances of this problem. All heuristics except HGA work well in terms of objective values for this problem relative to an individual delivery strategy, but they differ largely in terms of CPU time. Meta heuristics may generate a good or even optimal solution if running for a very long time, but we recommend using the Modified Sweep (MS) method or MS joint with TS. MS captures many important aspect of this IRP, consistently producing good solutions in minimal time.

For future work, it is interesting to develop multiple-product IRP, and current one-to-many network can be extended to many-to-many network.

Appendix

Minimize:

$$y = (a + cd_v)\gamma_v + \sum_{r \in S_v} h_r \left(\frac{0.5\mu_r}{\gamma_v} + z_\alpha \sigma_r \sqrt{\frac{1}{\gamma_v} + \frac{d_v}{p}} \right)$$

Let:

$$\begin{aligned}
&x = \gamma_v \\
&A = a + cd_v \\
&B = \textstyle\sum_{r\in S_v} 0.5 h_r \mu_r \\
&C = \textstyle\sum_{r\in S_v} 0.5 h_r z_\alpha \sigma_r \\
&D = d_v \,/\, p
\end{aligned}$$

Then the objective function becomes:

$$y = Ax + \frac{B}{x} + C\sqrt{\frac{1}{x} + D}$$

where *A, B, C*, and *D* are positive constants, *x* is an independent positive variable. We seek to find the value of *x* to minimize *y*.

$$\begin{aligned}
\frac{dy}{dx} &= A - \frac{B}{x^2} + \frac{C}{2}\frac{1}{\sqrt{\frac{1}{x}+D}}\left(-\frac{1}{x^2}\right) \\
&= A - \frac{B}{x^2} - \frac{C}{2x^2\sqrt{\frac{1}{x}+D}} \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\frac{d^2y}{dx^2} &= \frac{2B}{x^3} - \left[\frac{C}{2x^2}\left(-\frac{1}{2}\right)\left(\frac{1}{x}+D\right)^{-3/2}\left(-\frac{1}{x^2}\right) + \left(\frac{1}{x}+D\right)^{-1/2}\left(-\frac{C}{x^3}\right)\right] \\
&= \frac{2B}{x^3} - \left[\frac{C}{4x^4\left(\frac{1}{x}+D\right)^{3/2}} - \frac{C}{x^3\sqrt{\frac{1}{x}+D}}\right] \\
&= \frac{2B}{x^3} + \frac{-C + 4Cx\sqrt{\frac{1}{x}+D}}{4x^4\left(\frac{1}{x}+D\right)^{3/2}} \\
&>0
\end{aligned}$$

References

[1] Abdelmaguid, T.F., Dessouky, M.M. and Ordonez, F., "Heuristic approaches for the inventory-routing problem with backlogging," *Computers & Industrial Engineering*, 56, 4, 1519–1534 (2009).

[2] Andersson, H., Hoff, A., Christiansen, M., Hasle, G. and Lokketangen, A., "Industrial aspects and literature survey: combined inventory management and routing," *Computers & Operation Research*, 37, 9, 1515–1536 (2010).

[3] Anily, S. and Bramel, J., "An asymptotic 98.5% effective lower bound on fixed partition policies for the inventory-routing problem," *Discrete Applied Mathematics*, 145, 1, 22–39 (2004).

[4] Baita, F., Ukovich, W., Pesenti, R. and Favaretto, D., "Dynamic routing-and-inventory problems: A review," *Transportation Research Part A: Policy and Practice*, 32, 8, 585–598 (1998).

[5] Ballou, R.H., *Business Logistics: Supply Chain Management, 5th edition,* Prentice Hall, New Jersey (2003).

[6] Bard, J.F. and Nananukul, N., "A branch-and-price algorithm for an integrated production and inventory routing problem," *Computers & Operation Research*, 37, 12, 2202-2217 (2010).

[7] Bard, J.F., Jaillet, P. and Dror, M., "A decomposition approach to the inventory routing problem with satellite facilities," *Transportation Science*, 32, 2, 189–203 (1998).

[8] Chan, L., Federguen, A. and Simchi-Levi, D., "Probabilistic analyses and practical algorithms for inventory-routing models," *Operation Research*, 46, 1, 96–106 (2009).

[9] Chiang, W. and Russell, R.A., "Simulated annealing metaheuristics for the vehicle routing problem with time windows," *Annals of Operations Research*, 63, 1, 3–27 (1996).

[10] Ekasioglu, B., Vural, A. and Reisman, A., "The vehicle routing problems: A taxonomic review," *Computers & Industrial Engineering*, 57, 4, 1472–1483 (2009).

[11] Feo, T.A. and Resende, M.G.C., "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, 6, 2, 109–133 (1995).

[12] Glover, F., "Tabu search: A tutorial," *Interfaces*, 20, 4, 74–94 (1990).

[13] Goldberg, D.E., *Genetic Algorithms in Search Optimization and Machine Learning, 1st edition*, Addison-Wesley, Boston (1989)

[14] Javid, A.A. and Azard, N., "Incorporation location, routing and inventory decisions in supply chain network design," *Transportation Research Part E: Logistics and Transportation Review*, 46, 5, 582–597 (2010).

[15] Li, J., Chu, F. and Chen, H., "Performance evaluation of distribution strategies for the inventory routing problem," *European Journal of Operation Research*, 202, 2, 412–419 (2010).

[16] Lin, S., Lee, Z., Ying, K. and Lee, C., "Applying hybrid meta-heuristics for capacitated vehicle routing problem," *Expert Systems with Applications* 36, 2, 1505–1512 (2009).

[17] Liu, S. and Chen, J., "A heuristic method for the inventory routing and pricing problem in a supply chain," *Expert Systems with Applications*, 38, 3, 1447–1456 (2011).

[18] Moin, N., Salhi, S. and Aziz, N., "An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem," *International Journal of Production Economics*, 133, 1, 334–343 (2010).

[19] Shen, S. and Honda, M., "Incorporating lateral transfers of vehicles and inventory into an integrated replenishment and routing plan for a three-echelon supply chain," *Computers & Industrial Engineering*, 56, 2, 754–775 (2009).

[20] Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A. and Velasco, N., "A GRASP with evolutionary path relinking for the truck and trailer routing problem," *Computers & Operations Research*, 38, 9, 1319–1334 (2011).

[21] Viswanathan, S. and Mathur, K., "Integrating routing and inventory decision in one-warehouse multiretailer multiproduct distribution systems," *Management Science*, 43, 3, 294–312 (1997).

[22] Yu, Y., Chen, H. and Chu, F., "A new model and hybrid approach for large scale inventory routing problems," *European Journal of Operation Research* 189, 3, 1022–1040 (2008).

[23] Zachariadis, E., Tarantilis, C. and Kiranoudis, C., "An integrated local search method for inventory and routing decisions," *Expert Systems with Applications*, 36, 7, 10239– 10248 (2009).

[24] Zhao, Q., Wang, S. and Lai, K., "A partition approach to the inventory/routing problem," *European Journal of Operation Research* 177, 2, 786–802 (2007).

[25] Zhao, Q., Chen, S. and Zang, C., "Model and algorithm for inventory/routing decision in a three-echelon logistics system," *European Journal of Operation Research*, 191, 3, 623–635 (2008).