International Journal of Supply and Operations Management

IJSOM

August 2015, Volume 2, Issue 2, pp. 700-719
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com

Vehicle routing with time windows and customer selection for perishable goods

Masoud Rabbani[*][a], Mohammad Javad Ramezankhani[a], Hamed Farrokhi-Asl[b] and Amir Farshbaf-Geranmayeh[a]

[a] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

[b] School of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran

Abstract
Delivering perishable products to customers as soon as possible and with the minimum cost has been always a challenge for producers and has been emphasized over recent years as the global market becoming more competitive. In this study a multi-objective mix integer non-linear programming model is proposed to maximize both profits of a distributer and the total freshness of the several products to be delivered to customers with respect to their demands and with consideration of different soft time windows for each customer, heterogeneous distribution fleet and customer selection option for the distributer. The proposed model is solved with TH method. The two genetic algorithm and simulated annealing algorithm are used to solve large-sized problems. Finally, their results are compared to each other when the optimization software becomes unable of solution representation.

Keywords: Vehicle Routing Problem with Time Windows (VRPTW); Perishable Goods; Multi-Objective Programming; Genetic Algorithm; Simulated Annealing.

[*] Corresponding author email address: mrabbani@ut.ac.ir

1. Introduction

In today's modern and complicated system, it is very important for suppliers to distribute their products in an efficient manner. For many products the element of time plays an ordinary role in delivery procedure, but when it comes to the products that have short life span, the delivery procedure plays a significant roles. The term perishable goods is applied for the products that start deteriorating as soon as they are produced. The range of perishable goods is ranges from blood and its derivatives to daily newspapers that their reports are important to be read the day they are published and they actually expire the next day. Flowers which may wither even before being delivered and all kinds of food products are also considered perishable. In general there are two kind of deterioration. First, the product becomes outdated after a specified time like blood. Second, the products deteriorate by the time passage until they became unpleasant such as flowers, vegetables and foods.

With focusing on the perishable foods, we can consider various categorizations. Vegetables, fruits and prepared meals are labelled as highly perishable because they deteriorate significantly fast; besides, the costumers are sensitive about their freshness. That is, they want their orders to be delivered as fresh as possible. For instance, a costumer that orders a prepared lunch expects it to seem and taste fresh. Another example is a grocery shop which sells vegetables and it definitely wants them to be as fresh as if they have been brought from the farm. These factors highlights the significance of the delivery procedure significantly and delays in distribution or delivering low-quality products may impose a penalty on the supplier.

The above-mentioned factors are important for the suppliers to organize an efficient and effective schedule for their delivery trucks in order to maximize the freshness of the products delivered to the customers and to minimize the distributions costs at the same time. Therefore, they should have an operational distribution planning which specifies which vehicle should carry what kind of products to which customer at appropriate time to fulfill the objectives. This task needs to solve a vehicle routing problem (VRP) with consideration of time windows (VRPTW) and taking into account the perishability (VRPTW-P) of the products planned to be delivered. Whenever perishability becomes more important for a customer then it will have more impact on the vehicle routing planning.

It is noteworthy that the revenues of food suppliers are dependent on the condition and freshness of the products when they are received by the retailors. So, the planners should consider this fact that some customers request that their orders to be delivered in certain hours of the day, or they sometimes want their orders not be delivered in specific times of the day. For instance, hotels or restaurants may want their required food products for serving breakfast before the dawn every day. This signifies the role of VRP in raising revenues of suppliers and reducing their distribution costs, because they should consider the various ranges of customers from hotels, hospitals, prisons, universities, dormitories, and retailers while paying attention to certain demands of each costumer.

The scheduling should optimize the performance of the supplier's delivery plan. Obviously, an optimized daily routing plan has a considerable impact on a company that its core activity is distribution of foods. With regard to the increase of price of petrol, selecting an appropriate planning to optimize the daily vehicle routing has immediate and strategic impact on a performance of food suppliers. Moreover, there exists motivations for perishable food suppliers to apply such planning tool. One is that usually the planned routings are fixed and only small modification are made to them from day to day. The other practical reason is that if the main plan does not work desirably, the products of company loss their quality.

Due to the above mentioned facts and factors, an integrated well-disciplined schedule should be designed for their daily vehicle routings in a way that the supplier makes sure that the freshest products will be received to their customer and the distributions costs are effectively minimized. In other words, they need to apply VRP for their daily distribution scheduling. This calls for modelling a multi-objective planning system which aims at maximizing profit (revenue minus distribution and perishability costs), and to maximize freshness of the products to be delivered to a customer. These two objectives are in conflict with each other and an appropriate trade-off is needed here.

Following previous studies on literature of perishable goods', it seems that few studies examined the customer selection option. None of the recent articles considered that servicing a customer would result in excessive costs and reducing the total profit. There might be a faraway customer with a little amount of demands and delivering service to this customer might cost the distributer dearly and this is not preferable. And also, delivering products at the highest possible level of freshness considering the cost of distribution simultaneously is another subject which has been turned a blind eye to in the recent studies in the field of perishable products' distribution.

The rest of this article is organized as follows. A comprehensive relevant literature review in vehicle routing problem for perishable goods is reviewed in section 2. A mixed integer non-linear mathematical model for the problem is proposed in section 3 with respect to the assumptions considered in this paper. Some numerical examples are generated and solved by our model in section 4. Parameters tuning are presented in section5.Finally in the section 6, the conclusion of this work is summarized and some hints for future research are discussed.

2. Literature Review

There are so many articles discussing perishable goods most of which are in the field of pricing, inventory control and return policies to a retailer, but a few papers have focused on the distribution of perishable goods. It is worth mentioning that most of them are published in the recent years which shows the rising trend of the researchers to this subject. Hereafter, we will review some works that are close to our research. For more general VRP concept the two main references have been used widely in many studies (Golden, Raghavan, & Wasil, 2008; Toth & Vigo, 2001).

A stochastic VRPTW model has been constructed to determine the optimal routing plan, loads and departure time for each vehicle with consideration of stochastic and deterministic travel times for vehicles in two separate instances and also took perishability concept into account (Hsu, Hung, & Li, 2007). This is the base paper for so many related papers in this topic. A hybrid combination of genetic algorithms and construction heuristic for vehicle routing was proposed in the problem of scheduling and distributing ready-mixed concrete supply chain which is a perishable product (Naso, Surico, Turchiano, & Kaymak, 2007). A heuristic algorithm was developed for a vegetable distribution problem with hard time windows which includes vegetable perishability as a main factor (Osvald & Stirn, 2008). A multi-depot VRP for a real world distribution of fresh milk instance was published (Tarantilis & Kiranoudis, 2002).

From the solution point of view, some exact and heuristic algorithms have been reviewed in the field of VRPTW for perishable goods (Doerner, Gronalt, Hartl, Kiechle, & Reimann, 2008). A real case study in a Portuguese food distributer center was applied which is a very successful example for the impact of optimization programming for VRPTW-P (Pedro Amorim, Parragh, Sperandio, & Almada-Lobo, 2012) but some of the perishability-related assumptions are not approved in our opinion. Another successful application of VRPTW modeling in healthcare real case of picking up blood bags was presented in the Austrian Red Cross (Doerner et al., 2008) Integrated production scheduling with VRPTW with stochastic demands for perishable food products have also been conducted and a value deterioration has been added into the objective function as a deterioration rate (Chen, Hsueh, & Chang, 2009). A multi-objective mathematical formulation of VRP with hard time windows and homogeneous fleet has been proposed and an evolutionary algorithm is implemented for large-size problems (P Amorim & Almada-Lobo, 2014). Similarly, a bi-objective inventory routing problem for perishable goods by considering customer satisfaction level has been published recently which could be used as an extension for future studies (Rahimi, Baboli, & Rekik, 2014).

Table 1. comparison of the VRP related papers considering perishable products

Author	VRP type	Number of Objectives	Perishability	Products	Distribution Fleet	Customer Selection
(Hsu et al., 2007)	STW; TD	1	Stochastic	Single	Heterogeneous	No
(Osvald & Stirn, 2008)	HTW; TD	1	Deterministic	Single	Homogeneous	No
(Chen et al., 2009)	STW	1	Stochastic	Multiple	Homogeneous	No
(Pedro Amorim et al., 2012)	HTW	1	Deterministic	Single	Heterogeneous	No
(P Amorim & Almada-Lobo, 2014)	HTW	2	Deterministic	Multiple	Homogeneous	No
This paper	STW; TD	2	Stochastic	Multiple	Heterogeneous	Yes

STW: Soft Time Window, HTW: Hard Time Window; TD: Time Dependent travel time

In Table 1 our work is compared into the closest papers in the field of VRPTW-P with respect to some model's characteristics. It is worth mentioning that this paper is the first one to consider customer selection option considering perishability nature of the customer's demands.

3. Mathematical Model

Most of the works in the literature has the objectives such as minimizing the costs of distribution. The model presented below has an option of customer selection to maximize the revenue of selling products to a customer minus the loss of violating time window and distribution cost which is time dependent.

Some assumptions during the modeling process should be made which are listed below. Now, we present the mixed integer non-linear programming model with respect to the following assumptions:

- There is multiple products with specific rate of deterioration which is a function of travel time.

- The distribution fleet is heterogeneous with multiple vehicles and each vehicle has a particular capacity. Multiple products could be transferred by each vehicle and there is no limitation for the products type.
- Only one depot is considered in the problem.
- Once a vehicle leaves the depot, all the products are in their maximum freshness.
- Each customer has a specified demand of each product which has to be fulfilled if the customer is chosen to be serviced. Also there is a particular time window for each customer in which it should get service. The time windows could be violated as there is a penalty for this violation in the cost function. Only delays in time windows results penalty.
- The service time for each customer is a function of its demand.
- There is a customer selection option in the model which could select or neglect a customer because of increasing in costs.

The formulation and notation is based on the VRPTW formulation proposed before (Cordeau, Desaulniers, Desrosiers, Solomon, & Soumis, 2001). The indices, parameters and decision variables of the model is as follows:

Indices:
i, j : number of customers; $i, j = 1, 2, ..., m$
A : set of available routes between customers and depot; $(i, j) \in A$
n : number of products; $n = 1, 2, ..., N$
k : number of vehicles; $k = 1, 2, ..., K$
δ_i^+ : set of successors of i; $\delta_i^+ = \{ j \mid (i, j) \in A \}$
δ_i^- : set of predecessors of i; $\delta_i^- = \{ j \mid (j, i) \in A \}$
$0, m+1$: indices for depot
Parameters:
N : number of products
m : number of customers
t_{ij} : time of the route (i, j)
$c_{ij} = \alpha t_{ij}$: cost of the route (i, j)
d_i^n : customer i’s demand of product n
p^n : price of product n
a_i : customer i’s start of time window
b_i : customer i’s end of time window
g : goodwill loss for delay in the time window
λ^n : penalty for delay in the time window per product n’s unit
C^k : capacity of vehicle k
v^n : volume occupied per product n’s unit

r^n : deterioration rate of product n
s^n : service time of product n
lsl_i : shelf-life of the most perishable product of the customer i's total demand
$sl^n = \frac{100}{r^n}$: useful shelf-life of product n

Decision variables:
x_{ijk} : 1, if the route (i, j) is crossed by the vehicle k and 0 otherwise
z_i : 1, customer i is getting service and 0 otherwise
y_i^k : start time of servicing to the customer i by vehicle k
f_i^n : freshness of product n while the customer i is getting service
kk : total number of vehicles needed
w_i^n : 1, if a customer has a demand for product n and 0 otherwise

The mathematical model is represented below:

$$\begin{aligned} \max\ z_1 = & \sum_{n=1}^{N}\sum_{i=1}^{m} d_i^n p^n z_i - \sum_{k=1}^{K}\sum_{i=0}^{m}\sum_{j=0}^{m} c_{ij} x_{ijk} - g\sum_{i=1}^{m}\sum_{k=1}^{K}\left(\max\left\{y_i^k z_i^k, b_i\right\} - b_i\right) \\ & - \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{m} \lambda^n d_i^n \left(\max\left\{y_i^k z_i, b_i\right\} - b_i\right) \end{aligned} \tag{1}$$

$$\max\ z_2 = \sum_{n=1}^{N}\sum_{i=1}^{m} f_i^n \tag{2}$$

$$s.t. \sum_{k=1}^{K}\sum_{j\in\delta_i^+} x_{ijk} = z_i \qquad i = 1,\ldots,m \tag{3}$$

$$\sum_{j\in\delta_0^+}^{m} x_{0jk} \le 1 \qquad k = 1,\ldots,K \tag{4}$$

$$\sum_{i\in\delta_j^-}^{m} x_{ijk} - \sum_{i\in\delta_j^+}^{m} x_{jik} = 0 \qquad k = 1,\ldots,K \tag{5}$$

$$\sum_{i\in\delta_{n+1}^-}^{m} x_{i(\mathrm{n}+1)k} \le 1 \qquad k = 1,\ldots,K \tag{6}$$

$$y_i^k + \sum_{n=1}^{N} d_i^n s^n + t_{ij} - M\left(1 - x_{ijk}\right) \le y_j^k \qquad k = 1,\ldots,K; \forall (i, j) \in A \tag{7}$$

$$\sum_{n=1}^{N}\sum_{j\in\delta_i^+}^{m} d_i^n v^n x_{ijk} \le C^k \qquad k = 1,\ldots,K \tag{8}$$

$$\sum_{k=1}^{K}\sum_{i\in\delta_{n+1}^-}^{m} x_{i(\mathrm{n}+1)k} = kk \tag{9}$$

$$y_0^k + lsl_i - y_i^k \ge 0 \qquad k = 1,\ldots,K\ ,\ i = 1,\ldots,m \tag{10}$$

$$f_i^n \le \frac{y_0^k + sl^n - y_i^k + M\left(1-z_i\right)}{sl^n} \qquad k=1,...,K \;,\; i=1,...,m \;,\; n=1,...,N \tag{11}$$

$$f_i^n \le w_i^n z_i \qquad k=1,...,K \;,\; i=1,...,m \;,\; n=1,...,N \tag{12}$$

$$w_i^n \le d_i^n \le M w_i^n \tag{13}$$

$$x_{ijk} \in \{0,1\} \qquad k=1,...,K \;,\; i=0,...,m+1 \;,\; j=0,...,m+1 \tag{14}$$

$$z_i \in \{0,1\} \qquad i=1,...,m \tag{15}$$

$$y_i^k \ge 0 \qquad k=1,...,K \;,\; i=1,...,m \tag{16}$$

The first objective (1) maximizes the total profit. First term is the revenue of fulfilling selected customers' demand. Term two refers to the distribution costs and the third term shows the goodwill loss due to delay in servicing customers and the fourth term considers the goodwill loss due to delay in servicing customers in respect of each product's per unit. The second objective (2) maximizes the sum of products' freshness for the selected customers in the start time of servicing to the one. Constraints (3) ensure that each customer should receive service by at most one vehicle. Constraints (4) ensure that if a vehicle services any customer, it starts its route from the depot. Constraints (5) guarantee that a vehicle enters a customer then it should leave this customer. Constraints (6) state that a vehicle left the depot should return back to the depot. Constraints (7) ensure the consistency of time. Constraints (8) ensure that the capacity of each vehicle does not trespass from capacity limitations. Constraints (9) determine the number of vehicles used in general. Constraints (10) ensure that no customer will receive rancid products. Constraints (11) determine the freshness percent of each product received by each customer. Constraints (12) ensure that the freshness would not exceed its maximum amount which is 100% or 1. Constraints (13) determine whether a customer has a demand for a product or not. Constraints (14)-(16) are bounding constraints.

4. Methodology

To solve this multi-objective mixed integer non-linear programming problem an interactive solution procedure is introduced as TH method (Torabi & Hassini, 2008). TH method is a novel fuzzy approach which is applicable in multi-objective problems to find an efficient compromise solution. Several articles in the past years have affirmed that the TH method is an exceptional fuzzy approach which has computational advantages over other methods and also can provide efficient solutions based on the decision maker's preferences. TH method eases up the decision makers' doubt to select between the proposed solutions in an efficient frontier by providing one solution in the end. Furthermore, this method gives satisfying flexibility to generate different solutions in order to help the decision maker select the preferred and practical solution.

The main steps of the TH method in our problem can be summarized as follows:

Step 1. Determine the positive ideal solution (PIS) and negative ideal solution (NIS) for each objective function solving the corresponding MINLP model as follows:

$$Z_1^{PIS} = \max z_1 \qquad Z_1^{NIS} = \min z_1$$
$$Z_2^{PIS} = \max z_2 \qquad Z_2^{NIS} = \min z_2$$

It should be noted that determining the above ideal solutions needs solving four mixed integer non-linear program which could be time-consuming especially in large-sized problems. Hence, implementation of heuristics or estimating NISs without solving the corresponding problem is strongly suggested (Torabi & Hassini, 2009).

Step 2. Specify a linear membership function for each objective function as follows:

$$\mu_1 = \begin{cases} 1 & z_1 > Z_1^{PIS} \\ \dfrac{z_1 - Z_1^{NIS}}{Z_1^{PIS} - Z_1^{NIS}} & Z_1^{NIS} \le z_1 \le Z_1^{PIS} \\ 0 & z_1 < Z_1^{NIS} \end{cases} \qquad (17)$$

$$\mu_2 = \begin{cases} 1 & z_2 > Z_2^{PIS} \\ \dfrac{z_2 - Z_2^{NIS}}{Z_2^{PIS} - Z_2^{NIS}} & Z_2^{NIS} \le z_2 \le Z_2^{PIS} \\ 0 & z_2 < Z_2^{NIS} \end{cases} \qquad (18)$$

In fact, μ_1 and μ_2 represent the satisfaction degree of each objective function for the given route plan. Figure 1 represents the linear membership function for Z_1.

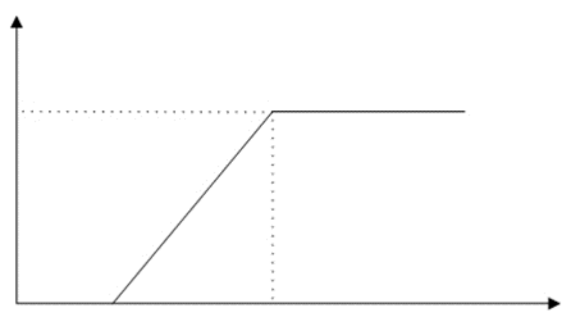

Figure 1. Linear membership function for z_1

Step 3. Convert the primary MINLP problem into a single-objective programming model using the following new formulation:

$$\max \ \lambda = \gamma\lambda_0 + (1-\gamma)\sum_h \theta_h \mu_h \qquad (19)$$

$$s.t. \ \lambda_0 \le \mu_h \qquad h = 1,2 \qquad (20)$$

$$\text{constraints (3) to (16)} \qquad (21)$$

$$\gamma \in [0,1] \qquad (22)$$

The μ_h and $\lambda_0 = \min_h \{\mu_h\}$ represent the satisfaction of h-th objective function and the minimum degree of objectives, respectively. This formulation has a new objective function combined of the lower bound for satisfaction degree of objectives (λ_0) and the weighted sum of the achievement degrees (μ_h) to ensure obtaining an adjustable balanced solution. Furthermore, θ_h and γ represent the relative importance of the h-th objective function and the importance of the lower bound of satisfaction, respectively. The θ_h values are determined by the decision maker based on one's opinion of the each objective function importance which can be derived using a multi-attribute decision making approach like AHP. The θ_h values are positive and $\sum_h \theta_h = 1$ should be satisfied. The proposed model is capable of taking decision maker's opinion into account for producing balanced and unbalanced solutions for the given problem through adjusting γ as a lower bound satisfaction preference parameter. The higher value of γ means maximizing the lower bound of satisfaction is more important which results in producing more balanced solutions. On the other hand, the lower value of γ means maximizing the sum of objective functions is more important and it results in high satisfaction degree for some objectives which will lead to more unbalanced solutions.

It should be noted that there is a correlation between γ and the range of θ_h's variation so there should be a limited rational interval for γ in which it could be determined for a given θ_h vector. For instance, if θ_hs change in a wide range then the corresponding γ should be determined as a small value (e.g. smaller than 0.3) because there is an explicit preference for the decision maker to get an unbalanced solution in the relative case.

Step 4. Determine appropriate values for γ and θ_h then solve the problem. If the decision maker is satisfied with the current solution, stop. Otherwise, consider another value for γ or θ_h and then go back to step 3.

To test our model some random examples has been generated and the two Genetic Algorithm (GA) and Simulated Annealing (SA) have been adopted to obtain reasonable results in the large-sized instances. The proposed model with abovementioned assumptions has been written in a GAMS 24.0 IDE coding language and it had produced optimal solutions for small-sized instances in a reasonable time. In the large-sized instances the GAMS IDE could not solve the problem due to NP-hard nature of the VRPTW-P problem. The results are obtained from a computer with a normal processing memory. The computer's specifications are as follows: Intel(R) Core(TM) i5-M480 @2.67 GHz 4 CPUs processor, 8192MB RAM and on a Windows 7 Home Premium 64-bit operating system.

4.1. Genetic Algorithm

Genetic algorithm has become popular as a tool to solve NP-hard optimization problems. Explaining the history, basic concepts and theoretical aspects of GA is not the aim of this paper and there is a lot of rich references for this purpose (Glover & Kochenberger, 2003). Only some general pseudo-code and some vital operators which are used in our solution algorithm are discussed. The general genetic algorithm pseudo-code is shown in the **Error! Reference source not found.**.

```
Create an initial random population of chromosomes;
while termination condition not satisfied do
  repeat
    evaluate the initial population;
    sort them;
    eliminate the excessive members
      if crossover condition satisfied then
      select parent chromosomes using roulette wheel selection algorithm;
      choose crossover parameters;
      perform crossover;
      save the crossover offspring population;
      if mutation condition satisfied then
      select parent chromosomes;
      choose mutation parameters;
      perform mutation;
      save the mutation offspring population;
    merge the initial, crossover and mutation populations;
  until sufficient offspring created;
endwhile
```

Figure 2. General GA pseudo-code

4.1.1. Chromosome Representation

In order to apply the GA to this particular problem, we need to construct algorithm to initialization of the algorithm. In our approach, a chromosome representing a set of vehicles and their customers is given by an integer string with the length of $+k$, where m is the number of customers and $k$is the number of vehicles available. A gene smaller/equal than m in a given chromosome indicates the original customer assigned to a vehicle which is the gene next greater than m in the chromosome, while the sequence of genes in the chromosome string indicates the order of visitation of customers. An example of a chromosome considering 7 customers and 3 vehicles resulting in a solution for the network given in **Error! Reference source not found.** as follows:

$$1\text{-}\ 8\text{-}\ 7\text{-}\ 2\text{-}\ 6\text{-}\ 5\text{-}\ 3\text{-}\ 9\text{-}\ 10\text{-}\ 4$$

$$L_1 = \{1\}$$

$$L_2 = \{7 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 3\}$$

$$L_3 = \varnothing$$

Figure 3. Chromosome representation

In this representation L_k represents the sequence of customers in route of vehicle k. In this example, customer 4 is eliminated and vehicle 3 has been remained unused.

4.1.2. Crossover

Ordered Crossover (OX) is used in our algorithm. In OX operator two crossover points are randomly selected from the parents' chromosomes to produce the offsprings. The two crossover points give a matching selection which is used to affect a cross through position by position exchange operations. The steps of this crossover are summarized and an example for this crossover is shown in **Error! Reference source not found.**.

1) Randomly select two cut points. This will result in a swathe of parents' genes. Select the swathe of parent 1 and copy them directly to the child 1's same indexes of the segment and do the same for parent 2 and child 2.
2) Look up for the parent 1's selected swathe genes in the parent 2 and eliminate the same genes. Do the same for parent 2's selected swathe and parent.
3) Copy any remaining parent 1's genes in the empty positions of the child 2 while keeping the same sequence of genes. Do the same for parent 2's genes and child 1's empty positions.

Parent 1: 1 3 | 4 5 | 2 7 6

Parent 2: 2 4 | 1 7 | 6 5 3

Child 1: 2 1 | 4 5 | 7 6 3

Child 2: 3 4 | 1 7 | 5 2 6

Figure 4. The OX crossover operator example

4.1.3. Mutation

We used three well-known mutation operators. The Insertion, Reversion and Swap operators have been used in a random Switch/Case concept in our algorithm. For further information the reader should study the work of Glover & Kochenberger (2003) in the case of unfamiliarity with the mentioned operators.

4.1.4. Selection

A Roulette Wheel selection algorithm has been used in our algorithm. In roulette wheel selection, parents are given a probability of being selected that is directly proportional to their fitness evaluation. A parents is then chosen randomly based on this probability and produce offspring. For further information refer to the Glover & Kochenberger (2003).

4.2. Simulated Annealing

Simulated annealing (SA) is a local search algorithm capable of escaping from local optima. Its ease of usage and convergence attributes and its use of hill-climbing moves to escape local optima have made it a popular technique over the past two decades. It is usually used in discrete optimization problems and to a lesser extent, continuous optimization problems. Explaining the history, basic concepts and theoretical aspects of SA is not the aim of this paper and there is a lot of rich references for that purpose (Eglese, 1990). The general structure of SA algorithm is shown in the Figure 5.

Create an initial random solution w;
Set the counter k to zero;
Select a temperature cooling schedule t_k;
Select an initial temperature $T = t_0$;
Select a repetition schedule, M_k, that defines the number of iterations executed at each temperature, t_k
while termination condition not satisfied **do**
  set repetition counter m to zero;
  repeat
    generate a solution w';
    calculate $\Delta E_{w,w'} = f(w') - f(w)$;
    if $\Delta E_{w,w'} \leq 0$ **then** $w' \rightarrow w$
    if $\Delta E_{w,w'} > 0$ **then** $w' \rightarrow w$ with the probability of $e^{-\frac{\Delta E_{w,w'}}{t_k}}$
    $m = m + 1$
  until $m = M_k$;
    $k = k + 1$;
endwhile
	Index:
	f : cost function
	w^* : best found solution
	w' : the proposed new solution
	W : the current value of search variable
	ΔE : energy level difference of W and w'
	k: algorithm's iteration counter
	m: repetition counter
	T : temperature

Figure 5. General SA pseudo-code

4.2.1. Annealing Schedule

The annealing schedule method to reach to the efficient energy level is a critical point in SA algorithm. We adopted equation (23) for our algorithm.

$$t_{k+1} = \alpha t_k \tag{23}$$

where t_k represents the temperature in the k-th iteration and α represents the temperature damping rate. It's been proven that $0.8 < \alpha < 0.99$ is an appropriate range for α to be selected from. The higher value of α would result in deeper search in the problem's solution set.

4.2.2. Neighbor Structure

We used three well-known neighbor creation operators. The Insertion, Reversion and Swap operators have been used in a random Switch/Case concept in our algorithm. For further information the reader should study the work of (Eglese, 1990).

5. Parameters Tuning

Both GA and SA is used to solve the problem. Both these two algorithms have several parameters which are all needed to be tuned properly to lead us to the best possible solution in a reasonable amount of time. GA has the initial population size (n_{pop}), mutation rate (p_m) and crossover rate (p_c) parameters to be tuned. Similarly, SA has the number of inner iterations (k), initial temperature in the inner iterations (t_0) and the temperature damping rate (α). The Taguchi design is used to design the required experiments and achieve the tuning of parameters' goal. Three levels are considered for each factor to be tuned and Minitab software is used to design experiments and analyze the results. For tuning parameters, a medium-size problem is solved considering the Taguchi method. The result following results of experiments and analysis are shown as follows:

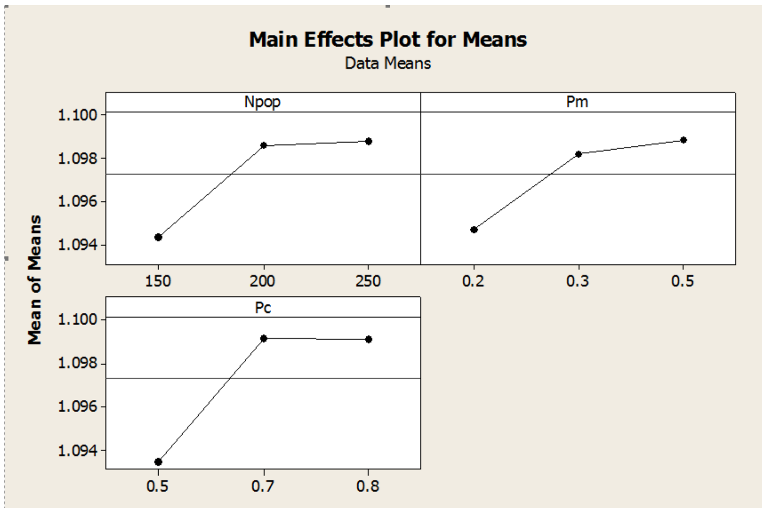

Figure 6. Analysis of Taguchi design for GA parameters

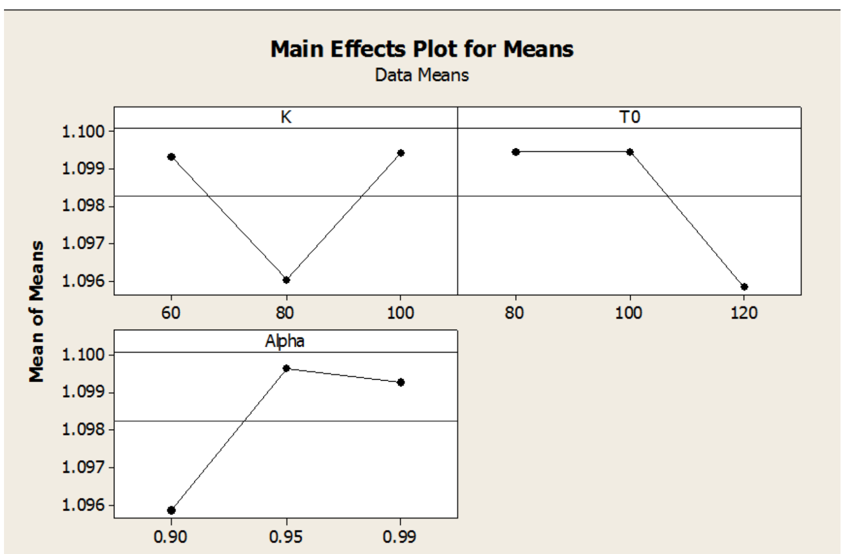

Figure 7. Analysis of Taguchi design for SA parameters

Table 2. Taguchi design of experiment and results for GA

Taguchi Design	Parameters			Objective Function
	n_{pop}	p_m	p_c	
Experiment 1	150	0.2	0.5	1.0848
Experiment 2	150	0.3	0.7	1.0990
Experiment 3	150	0.5	0.8	1.0994
Experiment 4	200	0.2	0.7	1.0995
Experiment 5	200	0.3	0.8	1.0982
Experiment 6	200	0.5	0.5	1.0981
Experiment 7	250	0.2	0.8	1.0999
Experiment 8	250	0.3	0.5	1.0975
Experiment 9	250	0.5	0.7	1.0991

Table 3. Taguchi design of experiment and results for SA

Taguchi Design	Parameters			Objective Function
	k	t_0	α	
Experiment 1	60	80	0.90	1.0993
Experiment 2	60	100	0.95	1.0994
Experiment 3	60	120	0.99	1.0993
Experiment 4	80	80	0.95	1.1000
Experiment 5	80	100	0.99	1.0994
Experiment 6	80	120	0.90	1.0887
Experiment 7	100	80	0.99	1.0991
Experiment 8	100	100	0.90	1.0996
Experiment 9	100	120	0.95	1.0996

Considering the analysis reports shown in Figures 6 and 7 the suitable parameters for both algorithms which are tuned are as follows:

Table 4. Parameter tuning results

Algorithm	Parameters		
	k	t_0	α
SA	60	100	0.99
	Parameters		
	n_{pop}	p_m	p_c
GA	200	0.3	0.8

6. Computational Experiments

In order to perform the computational study about the effect of the distribution different scenarios in our proposed model, we generated some instances based on the randomly generated parameters which are shown in the Table 5.

Table 5. The generated parameters

Parameter	Procedure
d_i^n	Uniformly integer randomized between [0 , 50]
s^n	Uniformly randomized between [0.01 , 0.1]
v^n	Uniformly integer randomized between [1 , 5]
λ^n	Uniformly integer randomized between [0 , 10]
p^n	Uniformly integer randomized between [30 , 100]
r^n	Uniformly integer randomized between [1 , 5]
C^k	Uniformly integer randomized between [mean of total demand , 1.5*mean of total demand]
X	Uniformly integer randomized between [0 , 250]
Y	Uniformly integer randomized between [0 , 250]
α	10
t_{ij}	$\sqrt{\left(X_i - X_j\right)^2 + \left(Y_i - Y_j\right)^2}$
g	100
a_i	Uniformly integer randomized between [0 , lsl_i -3]
b_i	Uniformly integer randomized between [a_i , a_i +40]

*X : longitude position of a customer; Y : latitude position of a customer

Twelve different size instances are generated as it is shown in Table 6. The parameters of GA and SA algorithms are also shown in Table 4.

Regarding of solutions obtained by GAMS coding, GA and SA algorithms which have been coded in MATLAB software and the proposed methodology to solve the instances would result in a comparison table which could help us to determine which one of the solution tools is more practical in which instance with respect to its size. We used $\gamma = 0.9$ as the TH method's parameter discussed before to solve P1 to P12. As it is shown in the **Error! Reference source not found.** the GAMS is not capable for solving a large-size problems and it cannot even handle P4 problem.

With the 600 iteration limitation, the GA produces better solutions comparing to SA, but when the iteration limitation increases to 1200 then SA represents better solutions. This is shown in the Figures 8-9. Although, the gap between the solutions obtained by algorithms and the solutions obtained by GAMS is not noticeable.

Table 6. Random generated problems specifications

Name	Number of Customers	Number of Vehicles	Number of Products
P1	6	3	1
P2	10	4	2
P3	15	7	3
P4	20	8	4
P5	25	10	5
P6	30	12	5
P7	35	14	6
P8	40	15	6
P9	48	17	7
P10	55	20	7
P11	62	22	8
P12	70	23	8

Table 7. Computational results

Name	GAMS		GA		
	0bjective	Time(s)	objective	Time(s)	gap
P1	1.085	89.85	1.085	48.23	0%
P2	1.102	226.42	1.102	49.89	0%
P3	1.101	312.22	1.097	52.45	0.36%
P4	-	-	1.098	68.12	-
P5	-	-	1.101	77.48	-
P6	-	-	1.101	89.68	-
P7	-	-	1.098	122.83	-
P8	-	-	1.099	142.25	-
P9	-	-	1.102	189.66	-
P10	-	-	1.109	221.12	-
P11	-	-	1.103	261.78	-
P12	-	-	Infeasible	321.01	-

Table 7. Continued

Name	SA (600 iteration)			SA (1200 iteration)		
	0bjective	Time(s)	gap	objective	Time(s)	gap
P1	1.085	47.22	0%	1.085	58.22	0%
P2	1.102	51.21	0%	1.102	59.13	0%
P3	0.989	54.45	11.32%	1.098	61.23	0.27%
P4	0.879	71.22	-	1.099	99.56	-
P5	0.955	80.30	-	1.102	109.90	-
P6	0.912	85.45	-	1.103	145.26	-
P7	0.857	117.19	-	1.102	188.67	-
P8	0.798	168.77	-	1.105	234.88	-
P9	1.023	206.46	-	1.102	298.11	-
P10	0.884	243.05	-	1.112	395.06	-
P11	0.986	274.40	-	1.107	436.14	-
P12	Infeasible	328.78	-	Infeasible	577.22	-

Two representations of suggested routing plans are shown in the Figures 10-11 for P6 and p10 problems respectively.

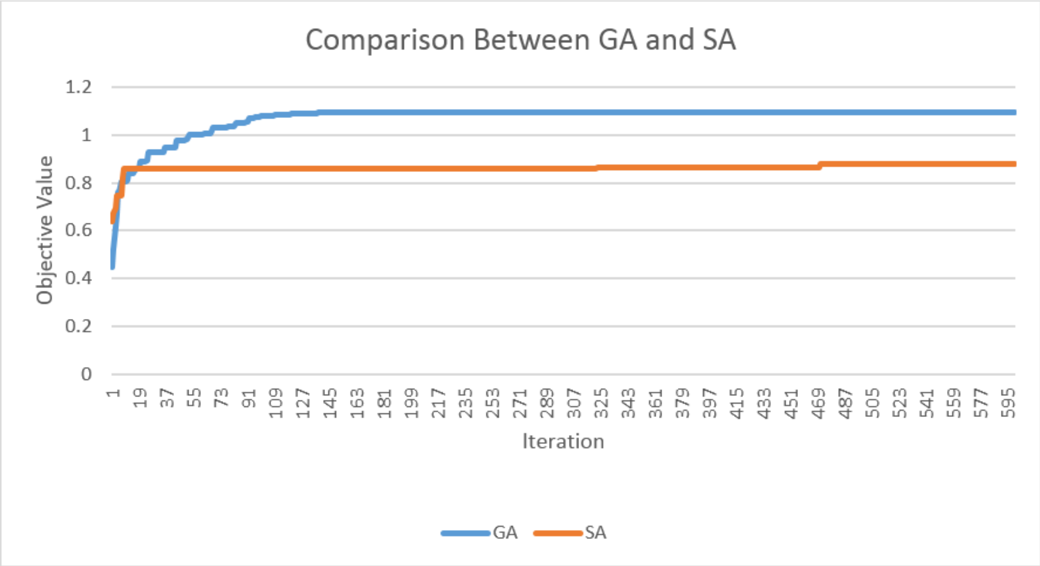

Figure 8. Comparison between GA and SA in P4 for 600 iterations

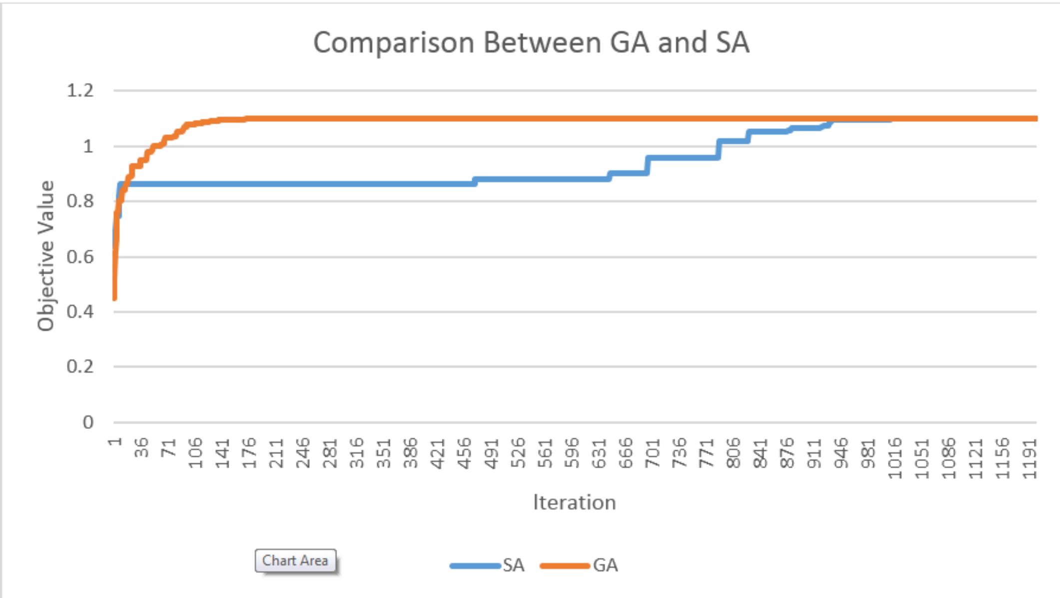

Figure 9. Comparison between GA and SA in P4 for 1200 iterations

As it is shown in P12, the generated problems does not lead to a feasible solution by our presented algorithms. This shows a real case data for a large-sized to test our proposed model.

The results also show that the GA is produced acceptable feasible solutions in large-size problems in a fair amount of time but SA's performance is better at the greater iterations and it produces near optimal solutions consuming more time and computational resources. It shows that our presented model and relating algorithms are performing successfully in large-size generated instances which could be adopted to solve real case problems in the future.

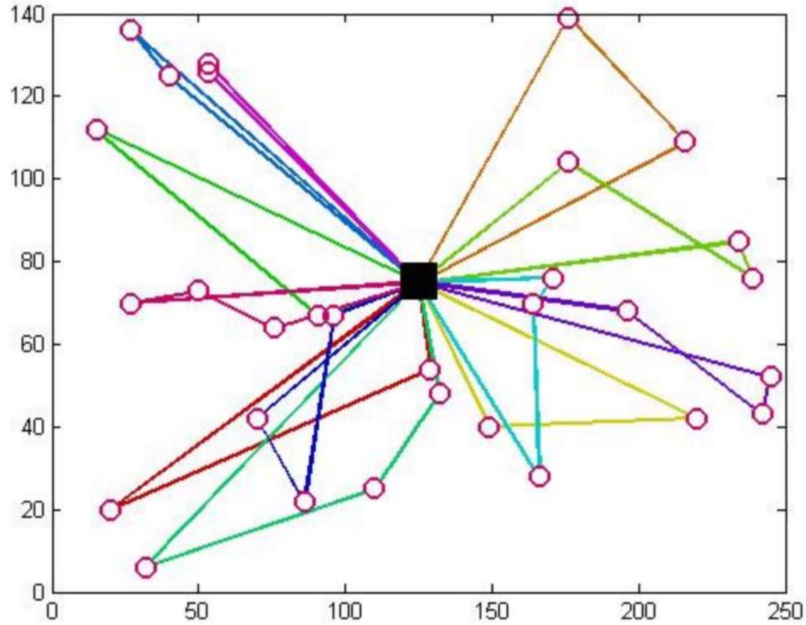

Figure 10. Routing plan representation for P6 problem

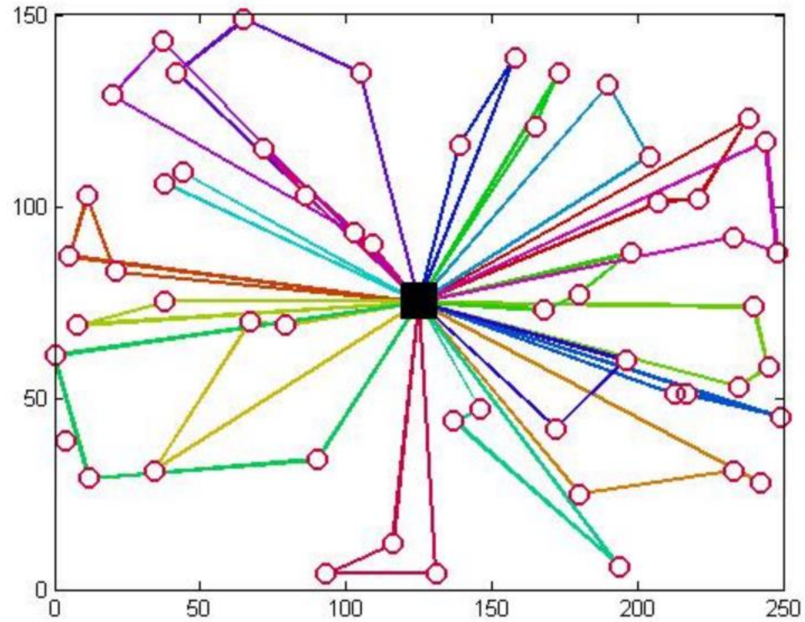

Figure 4. Routing plan representation for P10 problem

7. Conclusion

In this paper, a novel multi-objective mixed integer non-linear programming model was proposed for VRPTW considering perishability factors for perishable goods. The model had two different objectives: first objective aimed to maximize the profit of a distributer and the second one

intended to maximize the freshness of each customer's demands at the time of providing service to them. This kind of formulation could be used in any framework where the perishability becomes a critical issue for distributer. The proposed model converted to a single objective programming using TH method and two classic algorithms including Genetic Algorithm and Simulated Annealing Algorithm was used to solve large-sized problems. The proposed model in this paper had a special novelty which was the customer selection option which was not studied in the literature previously.
For future studies, some suggestions are described in the next sentences. The production phase before distribution can be studied more because the perishable goods' distribution can be extended to their production planning and the deterioration starts even before leaving the factory. Some extra VRP attributes can be added to this work such as multi-depot concept or cross docking center for perishable goods.

References

Amorim, P., & Almada-Lobo, B. (2014). The impact of food perishability issues in the vehicle routing problem. Computers & Industrial Engineering, 67, 223-233.

Amorim, P., Parragh, S. N., Sperandio, F., & Almada-Lobo, B. (2012). A rich vehicle routing problem dealing with perishable food: a case study. Top, 1-20.

Chen, H.-K., Hsueh, C.-F., & Chang, M.-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. Computers & Operations Research, 36(7), 2311-2319.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M. M., & Soumis, F. (2001). VRP with time windows. The vehicle routing problem, 9, 157-193.

Doerner, K. F., Gronalt, M., Hartl, R. F., Kiechle, G., & Reimann, M. (2008). Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows. Computers & Operations Research, 35(9), 3034-3048.

Eglese, R. (1990). Simulated annealing: a tool for operational research. European Journal of Operational Research, 46(3), 271-281.

Glover, F., & Kochenberger, G. A. (2003). Handbook of metaheuristics: Springer.

Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). The Vehicle Routing Problem: Latest Advances and New Challenges: latest advances and new challenges (Vol. 43): Springer.

Hsu, C.-I., Hung, S.-F., & Li, H.-C. (2007). Vehicle routing problem with time-windows for perishable food delivery. Journal of Food Engineering, 80(2), 465-475.

Naso, D., Surico, M., Turchiano, B., & Kaymak, U. (2007). Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. European Journal of Operational Research, 177(3), 2069-2099.

Osvald, A., & Stirn, L. Z. (2008). A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. Journal of Food Engineering, 85(2), 285-295.

Rahimi, M., Baboli, A., & Rekik, Y. (2014). A bi-objective inventory routing problem by considering customer satisfaction level in context of perishable product. Paper presented at the Computational Intelligence in Production and Logistics Systems (CIPLS), 2014 IEEE Symposium on.

Tarantilis, C., & Kiranoudis, C. (2002). Distribution of fresh meat. Journal of Food Engineering, 51(1), 85-91.

Torabi , S., & Hassini, E. (2009). Multi-site production planning integrating procurement and distribution plans in multi-echelon supply chains: an interactive fuzzy goal programming approach. International Journal of Production Research, 47(19), 5475-5499.

Torabi, S. A., & Hassini, E. (2008). An interactive possibilistic programming approach for multiple objective supply chain master planning. Fuzzy Sets and Systems, 159(2), 193-214.

Toth, P., & Vigo, D. (2001). The vehicle routing problem: Siam.