Computers and Chemical Engineering 95 (2016) 231–239

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A hybrid heuristic for the inventory routing problem under dynamic regional pricing

Farhad Etebari[a,*], Nooraddin Dabiri[b]

[a] Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[b] Engineering Faculty, Golestan University, Gorgan, Iran

ARTICLE INFO

Article history:
Received 16 February 2016
Received in revised form 6 September 2016
Accepted 25 September 2016
Available online 28 September 2016

Keywords:
Inventory routing problem (IRP)
Dynamic pricing
Regional and time-based pricing
Inventory routing under dynamic regional pricing problem (IRDRP)
Hybrid heuristic

ABSTRACT

The inventory routing problem (IRP) seeks to meet the demands of customers during consecutive time periods. Because of the geographical distribution of customers and variations in willingness to pay of the consumers in distinct locations and time, regional and time-based pricing are powerful ways to improve profitability. In this study, a quadratic mixed-integer programming model for single product, multi-period Inventory Routing under the dynamic regional pricing problem (IRDRP) has been proposed. A hybrid heuristic approach is developed to solve it. This algorithm comprises five phases: initialization, demand generation, demand adjustment, inventory routing, and neighborhood search, which are embedded in a simulated annealing framework. Experimental results indicate as the problem size increases, the difference between CPLEX and the proposed heuristic algorithm optimality gap exhibits an upward trend and that the heuristic outperforms CPLEX. A sensitivity analysis demonstrates that by intensifying the scarce capacity, approaching an optimal solution will be more difficult.

1. Introduction

The inventory routing problem (IRP) seeks to meet the demand of customers scattered in different geographical areas during consecutive times. The IRP manages the process of supplying one or more goods from one source to few destinations within the desired period with regard to the inventory and routing decisions (Campbell and Savelsbergh, 2004). In other words, the IRP integrates the two well-known problems of inventory and routing.

Where supply chain management addresses the supply decisions for increasing profitability of the system, demand management applies decisions for responding to fluctuations in consumer demand. One of the most prevalent tools for demand management is varying prices. Price differentiation is a mechanism for charging different consumers with different prices (Philips, 2005). Two common tactics for varying prices are regional and time-based pricing. These techniques are based on variations of the willingness of consumers to pay for receiving desired goods in distinct locations and times. Customers, particularly apparel and other seasonal-goods sectors, use markdown pricing to clear excess inventory before season end (Talluri, 2004). The demand for a commodity can be related to air temperature. For example, liquefied gas that used for home heating, worth more during the cold seasons and therefore, consumers have higher maximum willingness to pay for it during these seasons.

According to the dependency of demand and price, price variations affect demand and consequently, inventory and routing decisions in the IRPs. Simultaneous decision making for dynamic regional pricing, the rate of forwarded inventory to each customer and required routes for provided goods would increase the profitability of the supply chain. Inventory routing under the dynamic regional pricing problem (IRDRP) integrates these decisions to respond as necessary.

Since the inventory routing is NP-hard (Lenstra and Rinooy, 1981), this problem also falls under the set of hard problems. Heuristic and *meta*-heuristic algorithms should therefore be used to solve it. Since the IRDRP problems are more complex than the IRP, we will gain poor results by algorithms concentrating only on the inventory routing heuristics. Therefore, it is necessary to combine pricing optimization and the IRP heuristic algorithms for improving solutions.

Bell et al. (1983) and Federgrun and Zipkin (1984) conducted the first studies on the IRP in the mid-1980's. Bell et al. (1983) focused on gas distribution and determined routes, delivery quantities, visit times and used vehicles. Federgrun and Zipkin integrated problems of inventory and routing in a single period and single product

* Corresponding author.
E-mail address: etebari@qiau.ac.ir (F. Etebari).

http://dx.doi.org/10.1016/j.compchemeng.2016.09.018
0098-1354/

problem with random demand in customers, without considering the order costs. Their method for solving the problem consists of decomposing it into two problems: allocating inventory and routing vehicles (Federgrun and Zipkin, 1984).

Heuristic algorithms are the most widely established tools for solving IRP. Most of these include constructive and improvement heuristic solutions. Dror and Levy (1986) propose a two-phase algorithm based on the vehicle routing problem (VRP) initial solution. The route improvement heuristic constitutes the second phase. Abdelmaguid et al. (2009) propose a constructive and improvement heuristic approach for inventory routing with backlogging. Recently, Shao et al. (2015) develop a heuristic algorithm includes two phases. The construction phase is based on a rolling time algorithm and a greedy randomized adaptive search procedure; improvement phase relies upon neighborhood search. Cordeau et al. (2015) developed a three-phase heuristic for multiple-product IRP, including replenishment plans, delivery sequence, and routing. Vidovic et al. (2014) proposed a constructive heuristic algorithm with a variable neighborhood search. Incremental approach proposed by Singh et al. (2015) decomposes the set of customers into sub-problems that solved with a local search heuristic.

Branch-and-cut is a next approach incorporated by different authors (Archetti et al., 2007; Coelho and Laporte, 2013; Leandro and Gilbert, 2013, 2014). Lagrangian relaxation approach was developed to solve an IRP in crude oil transportation (Shen et al., 2011). Metaheuristic algorithms can also overcome the complexity of this problem. Abdelmaguid and Dessouky (2006) selected genetic algorithm to handle the complexity of the IRP. Shukla et al. (2013) employ a portfolio of evolutionary algorithms to solve this problem. Recently, Mjirda et al. (2014) developed a two-phase metaheuristic to solve the multi-product inventory routing problem. The first phase includes a metaheuristic to solve VRP; the second phase determines the amount of products to collect. Moin et al. (2011) developed another two phase hybrid genetic algorithm for solving this problem.

As mentioned, in recent decades, a large body of researches has been conducted in this area, and there are several comprehensive review of the IRP. Baita et al. (1998) study this problem as the Dynamic Routing and Inventory Problem (DPIR). Dynamicity in this problem is described as consecutive decisions in different time periods within planning horizon and as the effects of each decision on future decisions. Sarmiento and Nagi (1999) work on distribution-production systems. The next study was conducted by Moin and Salhi (2007) and the last study by Anderson et al. (2010). According to this study, the criteria of routing inventory models may include time, demand, topology, routing, inventory, composition, and size of fleet. Planning horizon may include a limited or unlimited period. Demand may be deterministic or stochastic. Topology of supply chain may be one-to-one, one-to-many, or many-to-many. Routes may be direct or multiple. Inventory could be fixed, stocked out, a lost sale or a backorder. Transportation fleet may be homogenous or nonhomogeneous and, finally, the fleet size may be single, multiple, or unconstrained.

Based on our surveys, the only study on pricing-routing-inventory is by Liu and Chen (2011). In this study, a *meta*-heuristic algorithm is proposed for solving this problem according to each customer's offering prices. The authors consider the single-period static pricing problem and do not incorporate the effects of supplying demand of a period from other periods in the price optimization algorithm. They ignore dynamic pricing during different periods of providing products.

Therefore, the main contribution of this study and the major difference between this study and the study by Liu and Chen (2011) is assuming dynamic pricing during consecutive periods for the same product. Incorporation of single-product, multi-period dynamic pricing with replenishment algorithm allows supplying demand of a period from other periods and the development of a demand-adjustment process, whereby restrictions are imposed on vehicle capacity. We develop an IRDRP algorithm for solving this problem and for determining related decision variables simultaneously.

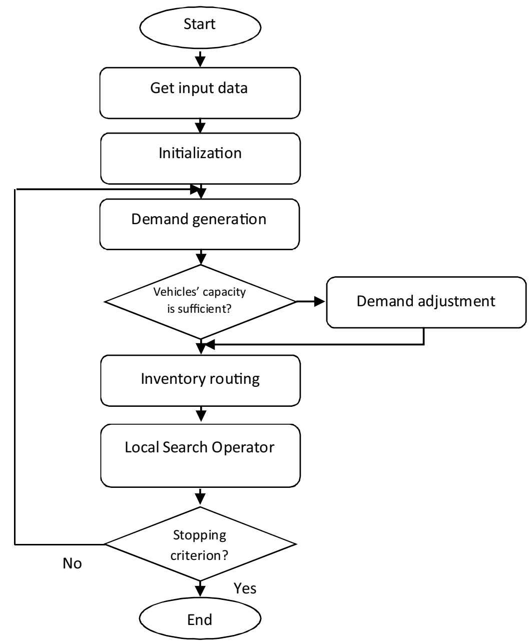


Fig. 1. The flowchart for proposed hybrid algorithm.

The rest of the article is as follows. A definition of the problem and its mathematical model is presented in section two. In section three, the proposed hybrid algorithm is presented. In the next section, numerical calculations required to determine the efficiency of the proposed algorithm and to compare it with exact solutions are considered. The last section consists in a summary and conclusion.

2. Model formulation

We assume that the problem includes N customers which are scattered in different geographical areas. Total number of time periods is equal to T. The purpose of this problem is to determine specific offering prices for each customer during each time period, inventory, and routes for sending the product from distributer to customers using rental transportation fleet with an upper bound on the number of available vehicles in order to maximize total revenue of supply chain. Linear price response function is used for modeling the relationship between the price and the demand of the products. The following notations are used to represents the mathematical programming formulation for the IRDRP.

Sets

$i, j, k, l = \{1, 2, \ldots, N\}$: Sets of all customers

$O = \{0\}$: Depot

$v = \{1, 2, \ldots, V\}$: Set of possible vehicles for rent

$t = \{1, 2, \ldots, T\}$: Time periods

Parameters

C_{ij}: Travel cost from customer i to customer j

q_v: Capacity of vehicle v

f_t: Fix cost for rent a vehicle which is used in period t

D_{it}: Total amount of potential demand of customer i in period t

m_{it}: Absolute value of the slope of demand function of customer i in period t

C_i: Capacity of customer i

h_i: Holding cost of each product unit in customer i in each period

Decision variables

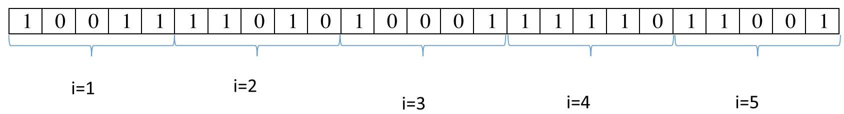


Fig. 2. A sample solution representation.

$x_{ijt}^{v}=\begin{cases}1 & \text{if vehicle } v \text{ goes directly from retailer } i \text{ to retailer } j \text{ at period } t\\ 0 & \text{otherwise}\end{cases}$

y_{ijt}^{v}: The quantity of product on vehicle v that goes directly from customer i to customer j at period t

p_{it}: The product offering price in customer i at period t

I_{it}: Inventory level in customer i at the: end of period t

The corresponding inventory-routing-pricing problem is modeled as follows:

$$Max\ \ f=\sum_{t=1}^{T}\Big(\sum_{i=1}^{N}p_{it}(D_{it}-m_ip_{it})-\sum_{j=1}^{N}\sum_{v=1}^{V}f_tx_{0jt}^{v}-\sum_{i=0}^{N}\sum_{\substack{j=0\\ i\neq j}}^{N}\sum_{v=1}^{V}c_{ij}x_{ijt}^{v}-\sum_{i=1}^{N}h_iI_{it}\Big) \tag{1}$$

$$S.t.\ \sum_{\substack{j=0\\ j\neq i}}^{N}x_{ijt}^{v}\ \leq 1;\forall i=0,1,\dots,N;t=1,2,\dots,T;v=1,2,\dots,V \tag{2}$$

$$\sum_{\substack{k=0\\ k\neq i}}^{N}x_{ikt}^{v}-\sum_{\substack{l=0\\ l\neq i}}^{N}x_{lit}^{v}=0;\forall i=0,1,\dots,N;t=1,2,\dots,T;v=1,2,\dots,V \tag{3}$$

$$y_{ijt}^{v}-q_vx_{ijt}^{v}\leq 0;i,j=0,1,\dots,N;i\neq j;t=1,2,\dots,T;v=1,2,\dots,V \tag{4}$$

$$\sum_{\substack{l=0\\ l\neq i}}^{N}y_{lit}^{v}-\sum_{\substack{k=0\\ k\neq i}}^{N}y_{ikt}^{v}\geq 0;\forall i=1,2,\dots,N;t=1,2,\dots,T;v=1,2,\dots,V \tag{5}$$

$$I_{it-1}-I_{it}+\sum_{v=1}^{V}\Big(\sum_{\substack{l=0\\ l\neq i}}^{N}y_{lit}^{v}-\sum_{\substack{k=0\\ k\neq i}}^{N}y_{ikt}^{v}\Big)=D_{it}-m_ip_{it};\forall i=1,2,\dots,N;t=1,2,\dots,T \tag{6}$$

$$I_{it}\leq C_i;\forall i=1,2,\dots,N;t=1,2,\dots,T \tag{7}$$

$$I_{it}\geq 0;\forall i=1,2,\dots,N;t=1,2,\dots,T \tag{8}$$

$$y_{ijt}^{v}\geq 0,p_{it}\geq 0\&x_{ijt}^{v}=0,1;i,j=0,1,\dots,N;i\neq j;t=1,2,\dots,T;v=1,2,\dots,V \tag{9}$$

The first equation indicates the objective function which seeks to maximize organization profit. The revenue function is a quadratic concave function leading to a convex maximization problem.

Constraint (2) prevents splitting the deliveries into customers in a time period, and Constraint (3) ensures that if vehicle v enters customer i during period t, it exits during the same period. This equality enforces route continuity. Constraint (4) serves two purposes. The first is to ensure that the amount handled between two locations will always be zero whenever there is no vehicle moving between them, and the second is to ensure that the amount transported is less than or equal to the vehicle's capacity. The left-hand term in constraint (5) indicates that the delivery amount for customer i must be positive. These constraints, along with other elements of the model, ensure that efficient solutions will not contain sub-tours. Constraint (6) balances the inventory levels at customers between periods. Constraint (7) indicates the capacity limitation of customers. Constraint (8) ensures that a shortage will not occur. According to definition of y_{ijt}^{v} Constraint (9) ensure loading amount of vehicles will not be negative and represents domain of x_{ijt}^{v}.

3. The proposed hybrid algorithm

Owing to the complexity of this problem (NP-hard), in order to produce a good solution with reasonable computation time, we should focus on heuristic and metaheuristic approaches. The proposed method in this paper is a heuristic algorithm embedded in a simulated annealing (SA) framework.

Fig. 1 presents the general process of the proposed algorithm. Input data includes routing, inventory holding, and pricing data. Customers' price response function during each period constitutes pricing data. First, we propose a structure to generate an initial solution as introduced in Section 3.1. A single-product-with-replenishment pricing heuristic generates customer product price and demand during different periods. This algorithm is presented in Section 3.2. If the demand at each customer is not relatively small compared to vehicle capacity, we will incorporate a demand adjustment process, as provided in Section 3.3. The obtained inventory routing problem will be solved by a heuristic solution which is explained in Section 3.4. Finally, according to Section 3.5, the neighborhood search approach under simulated annealing framework is used to local search operation.

3.1. Initialization

In order to determine the offering prices of products, possible time periods to replenish the inventory of customers must be specified. Therefore, the structure of the solution point is a binary vector in the size of $N*T$, where N is the number of customers and T is the number of time periods. The value 1 in this string indicates the possibility of visiting the desired customer and delivering goods during a specified period. Importing this solution to the pricing optimization algorithm ends in specifying the offering prices and demands of products. A sample solution representation with five customers and five time periods is shown in Fig. 2.

This solution demonstrates, for example, that the demands of customer 1 would be met by visiting transportation fleets during periods 1, 4 and 5.

3.2. Demand generation

In this section, we want to estimate the demand of different customers during each time period, based on the price response function. To this purpose, we refer to the literature on the pricing problem, and use an idea from the problem of single-product dynamic pricing with replenishment (Philips, 2005). The mathematical model of this problem is as follows:

$$\max \sum_{t=1}^{T} r(t, d(t)) - h_t x(t) - c_t y(t) \tag{10}$$

$$s.t.\ x(t) = x(t-1) - d(t) + y(t),\ t = 1, 2, \ldots, T \tag{11}$$

$$d(t), x(t), y(t) \geq 0,\ t = 1, 2, \ldots, T \tag{12}$$

In the above model, $r(t, d(t))$ represents customers' revenue during period t, based on the demand function; $d(t)$, h_t is a unit holding cost in period t, $x(t)$ denotes end-of-period inventory that can be replenished during offering horizon, c_t indicates all costs that a firm should pay for replenishment a unit inventory during period t; and, finally, $y(t)$ indicates the order amount during period t. The Eq. (10) indicates the objective function, which seeks to maximize the organization profit. Constraint (11) represents the constraint of inventory balance in different periods and Constraint (12) indicates the sign constraints. An efficient heuristic algorithm has been proposed to solve this problem. The parameter γ_{st} is defined as below:

$$\gamma_{st} = c_s + \sum_{k=s}^{t-1} h_k \tag{13}$$

In this equation, γ_{st} indicates the cost of responding to the demand of period t through supplying the goods in period s. This parameter includes the costs of supplying goods in a given period and total holding cost up to the final period. The cost of supplying goods in this problem includes transportation costs. In order to estimate the transportation costs for a customer in a particular period, the problem of the traveling salesman is solved in two modes: with and without the customer. We consider the difference as the estimated transportation cost. Customers which related element in the solution is 1 in the specified time period will be included in this tour. Suppose that s_{it} indicates the customer's number, which is planned to meet in repetition i in period t. So, $S_\tau = (s_{1t}, \ldots, s_{i-1,t}, s_{it}, s_{i+1,t}, \ldots, s_{Nt})$ indicates the meeting order of N customers in period t. Now, consider the $S_\tau^i = (s_{1t}, \ldots, s_{i-1,t}, s_{i+1,t}, \ldots, s_{Nt})$ as a new tour, in which customer i is removed from S_τ tour. If the cost of S_τ tour is represented by $TC(S_\tau)$, then the cost of supplying the desired goods of customer i is estimated by $TC(S_\tau) - TC(S_\tau^i)$.

Goods are supplied during a period in response to the demand of that period and of the following periods. Then we have:

$$\gamma^*(t) = \min_{s \leq t}\{\gamma_{st}\} \tag{14}$$

In which, $\gamma^*(t)$ indicates the minimum cost in order to supply the need of period t. Finally, the demand amount of each period is calculated by including the marginal revenue equal to this minimum cost, using the below equation:

$$J(t, d^*(t)) = \gamma^*(t);\ t = 1, 2, \ldots, T \tag{15}$$

$J(t, d(t))$ indicates the marginal revenue. $\gamma^*(t)$ has the interpretation as the lowest marginal cost for inventory. The optimality condition requires that the marginal revenue should equal the marginal cost of inventory in each period. In the case of violation of this rule, reallocating sales from a period of low marginal revenue to a period of high marginal revenue can increase revenue. .It is clear that, according to relationship between price and demand in each period, after specifying the demand amount, product price will be determined.

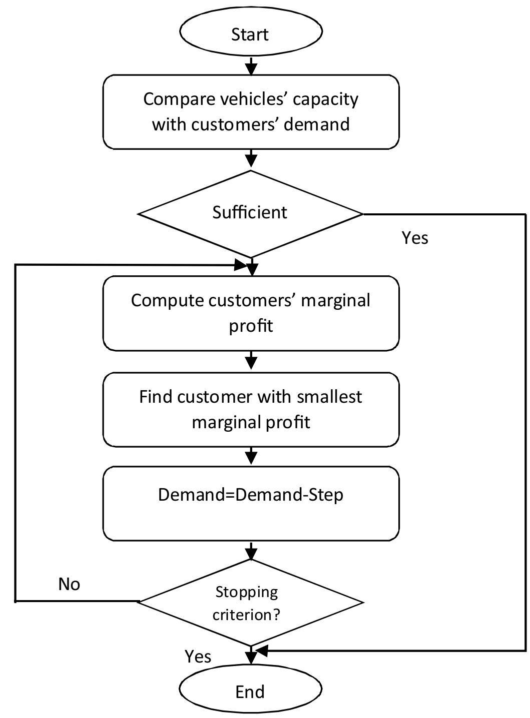


Fig. 3. Demand adjustment process.

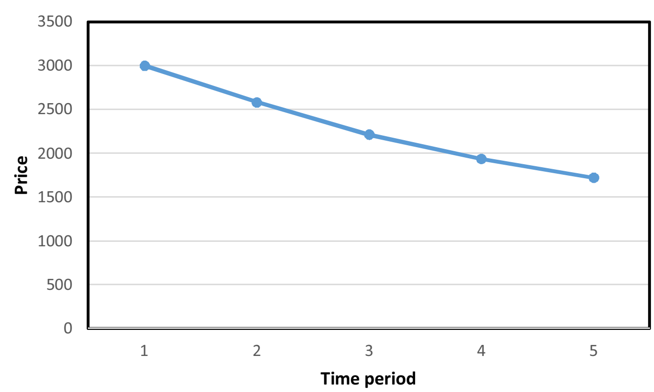


Fig. 4. Prices as a function of time periods prior to the end of the season.

3.3. Demand adjustment

If demand at each customer is not relatively small compared to the vehicles' capacity, customer demand should be adjusted. The demand adjustment process is presented in Fig. 3. At the beginning, the marginal profit of each customer at current demand level is calculated. Thereafter, these values are sorted, and the customer with minimal marginal value is specified. The demand level of the selected customer will be decreased in the step size. The accuracy and time length of the algorithm depends on the steps taken. Stopping criteria include the capacity of vehicles' capacity to satisfy customer demand. This process continues until the termination criteria are met.

3.4. Inventory-routing problem

In this stage, the problem will be reduced to the simple IRP. In this study, we have applied the heuristic algorithm proposed by Abdelmaguid et al. (2009) in order to solve the inventory-routing problem.

3.5. Local search operator

The heuristic algorithm explores the solution space by moving from one solution point to its neighbor, under the SA framework. This neighborhood search changes the delivery strategy by canceling some replenishment dates for various customers and incorporating new delivery dates. This process is implemented for μ percent of total time periods. In this regard, μ percent of each solution element is selected and its value changes according to the binary nature of these elements.

The SA algorithm was first introduced by Metropolis et al. (1953) in 1953. Our proposed algorithm is an SA-based heuristic. This algorithm is motivated by the physical annealing process in solids. An initial solution is generated, followed by a neighborhood solution created using a local search mechanism. If the new solution is better than the current one, the new solution is accepted; otherwise, the neighborhood solution is accepted with the probability $\exp(-{}^{\Delta}\!/_{KT})$. Δ is the objective difference between the new and current solution, k is the Boltzmann constant, and T is current temperature.

4. Computational analysis

The proposed algorithm was implemented in MATLAB and computational tests were run on a computer with a 2.50 GHZ processor and 3.00 GB RAM. The 64-bit version of the mathematical modeling suite GAMS 24.1.2 and the solver CPLEX was employed to solve the mathematical model. The obtained lowers and upper bounds of the mathematical model served as a benchmark for evaluating solution quality of the proposed algorithm.

4.1. Test problems generation

Computational tests were performed on a set of 50 instances. This data set is characterized by the number of customers and the number of time periods. These instances were generated from Abdelmaguid et al. (2009) by importing linear price response functions.

Fifty test problems are classified into seven sets, including first problem set with five customers and five to seven time periods, the second with ten customers and five to seven time periods, the third problem set with 15 customers and five to seven time periods, the forth problem set with 20 customers and seven time periods, and all of last three problem set performed with five time periods and 40, 70 and 100 customers.

Some goods, such as apparel goods, perishable foods, and liquefied gas, are seasonal which their values decrease once the season is over. Consumers price sensitivity increase for these goods during offering horizon. We set the slope of the linear price response function in the first period equal to 0.1 and increase it during the next periods, for the first data set. Fig. 6 shows evolution of prices for a particular customer in all time periods.

As expected, Fig. 4 shows markdown pricing according to increasing consumers price sensitivity.

4.2. Numerical analysis

In order to do better analysis we define instance sizes as number of customers multiplied by period numbers. We set the time limit of the CPLEX solver to 0.5 h for instances with sizes of 25 and 35, 2.0 h for instances with size of 50 and 70, and also, 3.0 h for instances with sizes 75, 105, and 140, finally 12.0 h for instances with sizes 200, 350, and 500. Table 1, shows the results of CPLEX and IRDRP algorithms for 50 instances.

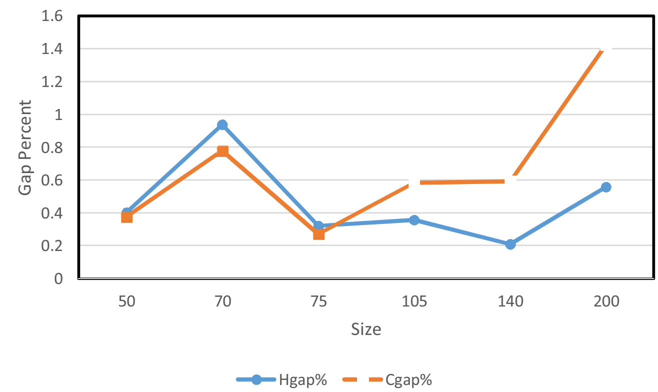


Fig. 5. CPLEX and IRDRP gap percent with respect to instances sizes.

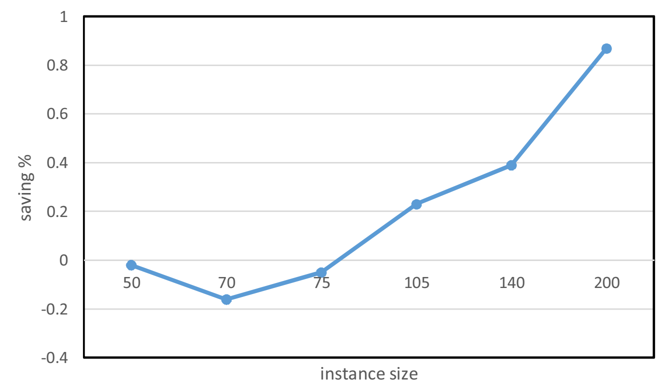


Fig. 6. Saving% of IRDRP against CPLEX with respect to instances sizes.

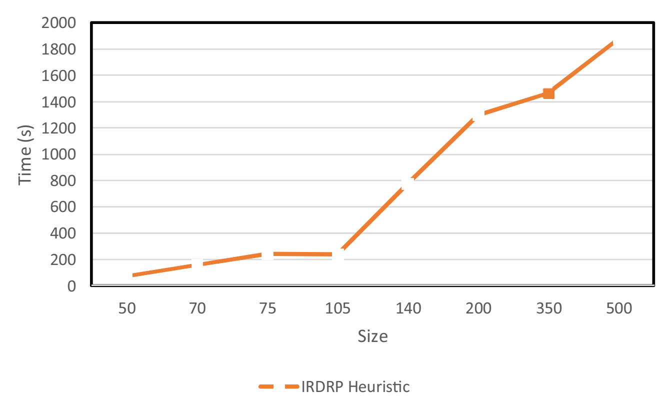


Fig. 7. Required time of IRDRP algorithm with respect to instances sizes.

In Table 1, we report the lower and upper bounds obtained by the CPLEX solver, the best results, averages and standard deviations obtained by the IRDRP heuristic, the deviation between the CPLEX lower and upper bounds (Cgap), the deviation between the CPLEX upper bounds and the IRDRP heuristic best feasible results (Hgap), and the IRDRP algorithm computational time. Last column determines saving percent of IRDRP algorithm against solutions obtained by the CPLEX solver. The indicators Cgap, Hgap and saving percentages are calculated by Eqs. (16)–(18), consequently.

$$Cgap\% = 100 \times \frac{Upper\ bound - Lower\ bound}{Upper\ bound} \quad (16)$$

$$Hgap\% = 100 \times \frac{Upper\ bound - IRDP\ result}{Upper\ bound} \quad (17)$$

$$saving\% = Cgap\% - Hgap\% \quad (18)$$

Table 1
Comparison between CPLEX and IRDRP algorithm results.

Test #	Customer	Period	Serial	Size	No. of variables	CPLEX			IRDRP				Hgap%	Saving%
						LB	UB	Cgap%	Best	Ave.	S.D.	Time(s)		
1	5	5	1	25	1140	15053	15117	0.43	14992	14979	21.5	13	0.82	−0.40
2	5	5	2	25	1140	12629	12665	0.29	12569	12510	45.9	9	0.75	−0.46
3	5	5	3	25	1140	15192	15244	0.34	15044	15037	7.8	15	1.31	−0.97
4	5	5	4	25	1140	20886	20939	0.25	20780	20758	14.9	11	0.75	−0.50
5	5	5	5	25	1140	9936	9990	0.55	9865	9853	10.1	9	1.25	−0.71
6	5	7	1	35	1596	16082	16174	0.57	15985	15958	34.9	17	1.16	−0.60
7	5	7	2	35	1596	17656	17686	0.17	17423	17383	26.6	15	1.48	−1.31
8	5	7	3	35	1596	17065	17157	0.53	16947	16939	4.9	19	1.22	−0.69
9	5	7	4	35	1596	18658	18761	0.55	18105	18063	40.8	14	3.49	−2.96
10	5	7	5	35	1596	18202	18312	0.60	17834	17670	93.7	12	2.6	−2.02
11	10	5	1	50	6160	65785	66034	0.38	65781	65770	7.2	73	0.38	−0.01
12	10	5	2	50	6160	44489	44684	0.44	44474	44463	8.3	92	0.47	−0.35
13	10	5	3	50	6160	56795	56983	0.33	56760	56740	26.8	60	0.39	−0.06
14	10	5	4	50	6160	58487	58714	0.39	58499	58495	4.3	81	0.36	0.02
15	10	5	5	50	6160	51181	51360	0.35	51157	51149	7.8	53	0.39	−0.04
16	10	7	1	70	8624	59416	60138	1.2	59279	59187	78.4	101	1.42	−0.22
17	10	7	2	70	8624	67712	68062	0.51	67480	67462	25.0	219	0.85	−0.34
18	10	7	3	70	8624	71933	72689	1.04	71864	71773	76.8	165	1.13	−0.09
19	10	7	4	70	8624	77141	77585	0.57	77074	77047	33.2	136	0.65	−0.08
20	10	7	5	70	8624	67514	67893	0.56	67478	67469	7.3	177	0.61	−0.05
21	15	5	1	75	12960	108150	108534	0.35	108145	108087	38.6	324	0.35	−0.00
22	15	5	2	75	12960	140986	141338	0.25	141042	141023	16.2	159	0.20	0.04
23	15	5	3	75	12960	146833	147194	0.24	146894	146878	16.9	181	0.20	0.04
24	15	5	4	75	12960	136950	137317	0.27	137012	136976	36.0	438	0.22	0.04
25	15	5	5	75	12960	150607	150973	0.24	150047	149957	97.5	116	0.61	−0.37
26	15	7	1	105	18144	213582	214313	0.34	213916	213883	28.6	276	0.18	0.15
27	15	7	2	105	18144	52076	52556	0.91	52171	52157	14.8	339	0.73	0.18
28	15	7	3	105	18144	174619	175611	0.56	175236	175218	16.0	118	0.21	0.35
29	15	7	4	105	18144	170001	170933	0.55	170391	170362	23.4	272	0.31	0.22
30	15	7	5	105	18144	145556	146358	0.55	145853	145777	55.2	182	0.34	0.20
31	20	7	1	140	31164	243980	245282	0.53	244744	244688	43.0	437	0.21	0.31
32	20	7	2	140	31164	267239	268607	0.51	268055	267957	71.4	415	0.20	0.30
33	20	7	3	140	31164	246074	248010	0.78	247445	247356	83.1	975	0.22	0.55
34	20	7	4	140	31164	315587	316918	0.42	316350	316276	46.1	1065	0.17	0.24
35	20	7	5	140	31164	209114	210623	0.72	210153	210103	33.3	991	0.22	0.49
36	40	5	1	200	118080	350284	353250	0.84	352485	351998	549	1463	0.21	0.62
37	40	5	2	200	118080	352035	354885	0.80	353930	353533	287	1251	0.26	0.53
38	40	5	3	200	118080	381026	388324	1.88	386487	386126	432	1430	0.47	1.43
39	40	5	4	200	118080	323390	326431	0.93	323862	323568	283	928	0.78	0.14
40	40	5	5	200	118080	336462	345666	2.66	342072	341533	637	1427	1.03	1.66
41	70	5	1	350	353280	–	628894	–	625142	623962	897	1665	0.59	–
42	70	5	2	350	353280	–	708793	–	704936	704435	331	1008	0.54	–
43	70	5	3	350	353280	–	648447	–	636593	633824	1696	1253	1.82	–
44	70	5	4	350	353280	–	607179	–	603539	601199	4229	1474	2.22	–
45	70	5	5	350	353280	–	654345	–	608965	607959	795	1921	0.67	–
46	100	5	1	500	715080	–	963935	–	958397	958368	64	1256	0.57	–
47	100	5	2	500	715080	–	944475	–	937198	936117	604	1960	0.77	–
48	100	5	3	500	715080	–	963935	–	887188	887083	80	1903	0.47	–
49	100	5	4	500	715080	–	944475	–	881237	880644	384	2305	0.67	–
50	100	5	5	500	715080	–	891396	–	982307	982075	371	1976	0.47	–

Results indicate that, for instances with sizes of 25 and 35, the exact method provides better results. The average gaps are 0.61 and 1.51 percent, respectively, providing better than saving average for these sizes. When the instance sizes increase to 50, 70, and 75, these averages are very close. By increasing sizes to 140 and more, the IRDRP algorithm outperforms the CPLEX and better results are achieved in limited times. Therefore we can conclude that proposed algorithm outperforms exact algorithm implemented by CPLEX, especially in practical problem sizes.

Fig. 5 compares the average gap and savings percent of CPLEX and IRDRP algorithms. We see that, in small size problems, CPLEX outperforms IRDRP algorithm. But as demonstrated in Fig. 6 when the size of problems increases, the difference between two indicators (i.e. saving%) exhibits an upward trend, and IRDRP outperforms CPLEX. Fig. 7 demonstrates effect of the scale of the problem on the required time for proposed algorithm.

Table 2
Comparison between IRDRP and IRSP algorithms results.

Customer	Period	Serial	IRDRP	IRSP
			Profit	Profit
5	5	1	14992	14698
5	5	2	12569	12094
5	5	3	15044	14798
5	5	4	20780	20160
5	5	5	9865	9461
10	5	1	65781	58852
10	5	2	44474	40223
10	5	3	56760	48667
10	5	4	58499	53185
10	5	5	51157	44779

Table 3
Sensitivity analysis by considering different scale factors for vehicles' capacities.

test #	Customer	Period	Serial	Scale factor	CPLEX			HHA		
					LB	UB	Cgap%	Profit	Time(s)	Hgap%
1	10	5	1	0.8	62571	62751	0.28	62571	71	0.28
2	10	5	1	1	65471	65627	0.23	65487	59	0.21
3	10	5	1	1.2	65827	65975	0.22	65835	58	0.21
4	15	5	1	0.8	106628	106968	0.31	106520	269	0.41
5	15	5	1	1	108229	108496	0.24	108171	339	0.29
6	15	5	1	1.2	108244	108513	0.24	108247	311	0.24
7	20	7	1	0.8	241087	241979	0.36	241225	476	0.31
8	20	7	1	1	244479	245308	0.33	244881	752	0.17
9	20	7	1	1.2	244415	245333	0.37	244903	916	0.17

Table 4
Sensitivity analysis by considering different types of vehicles.

Customer	Period	Serial	Scenario	CPLEX			HHA		
				LB	UB	Cgap%	Profit	Hgap%	Saving%
5	5	1	1	10571	10571	0	10111	4.35	−4.35
5	5	2	2	9779	9837	0.58	9116	7.32	−6.77
5	5	3	3	9415	9452	0.38	9074	3.99	−3.61
10	5	1	1	40561	41716	2.76	40522	2.86	−0.09
10	5	2	2	39942	40290	0.86	39345	2.34	−1.49
10	5	3	3	38734	39214	1.22	37843	3.49	−2.30

4.3. Dynamic pricing vs. static pricing

In this section we ignore dynamic pricing and determines same prices for each customer during offering horizon. Inventory routing under static pricing (IRSP) heuristic algorithm inspired from Liu and Chen (2011). This algorithm includes two main phases which are initial solution generation and improvement steps. During first step, we set products offering prices equal to zero and calculate the marginal revenue of each customer during offering horizon. We find the customer with the maximal marginal revenue. If $\max_{r,t}\{MR(r,t)\} > 0$, then we increment the price of the specified customer. $MR(r,t)$ indicates marginal revenue of the customer r during period t. Otherwise, if $\max_{r,t}\{MR(r,t)\} \le 0$, we stop. Abdelmaguid et al. (2009) heuristic algorithm is used for solving IRP and pricing improvement is done based on the method proposed by Liu and Chen (2011).

The results of the IRDRP and IRSP algorithms for small size problems are presented in Table 2. It shows that the IRDRP algorithm significantly outperformed the static pricing approach. These results reveal the necessity of applying dynamic pricing optimization methods in the heuristic algorithm.

4.4. Impacts of vehicles' capacity

This analysis is done by evaluating the proposed algorithm performance by scarce capacity. We use a parameter α to scale all vehicle capacities, where $\alpha = 1$ corresponds to the original baseline case.

Here, we assumed vehicle capacities to be adequate for delivering the required goods during each period. In this section, we test different vehicle capacities for evaluating IRDRP algorithm performance. We used a parameter α to scale vehicle capacities, where $\alpha = 1$ corresponds to the original baseline case.

Table 3 summarizes the objective function obtained under different vehicle capacities. For example, the first row in the table represents the case $c_{0.8} = 0.8 \times 450$.

In these instances, there are two solutions: CPLEX outperforms the IRDRP algorithm and, in seven instances, the IRDRP algorithm achieves tighter optimality gap. The CPLEX time limit is similar to the mentioned limitations in Table 1.

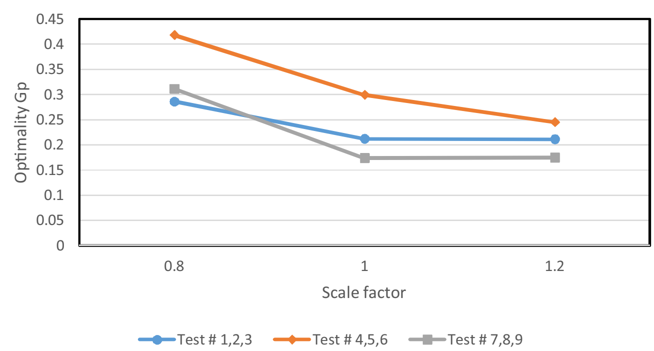


Fig. 8. IRDRP heuristic optimality gap for different scale factors in three sizes. (a)Test instances number 1–3. (b)Test instances number 4–6. (c) Test instances number 7–9.

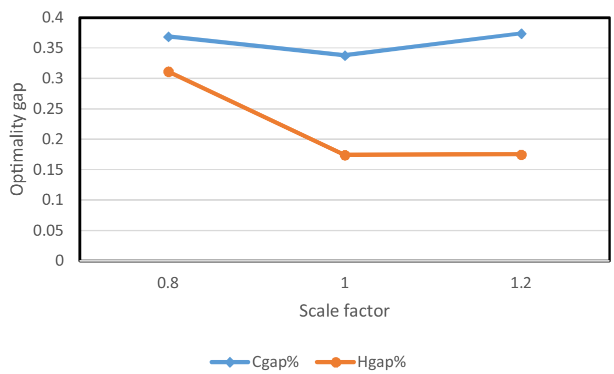
Fig. 8 illustrates the optimality gaps under different scale factors. As scarce capacity settings intensify, approaching the optimal solution becomes more difficult, and we observe an upward trend in percentage of savings. This figure represents an available scarcity increase and optimality gap increase in all problem sizes.

Fig. 9 presents an optimality gap with respect to scale factor parameter. Figs. 9a and c illustrate the downward interval between the optimality gap of CPLEX and IRSRP algorithm as a decreasing scale factor. Fig. 8b shows the simultaneous increase in optimality gap in both algorithms.

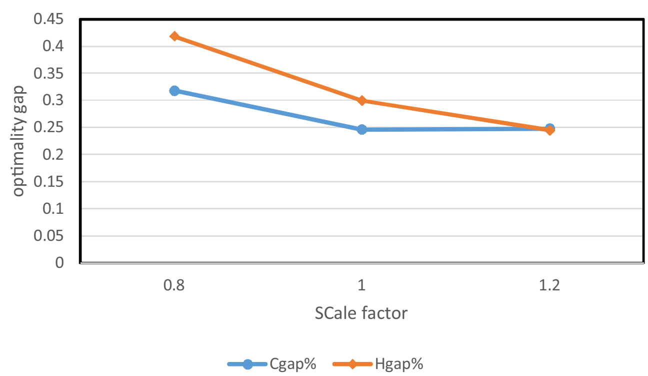
4.5. Impacts of vehicle types

The last sensitivity analysis in done based on vehicle types. We design three experiment which the supplier has different vehicle types by different capacity and fixed cost. It is assumed that by increasing the vehicle's capacity, its fixed cost will be increased. Table 4 represents the objective function of two algorithms under three scenario of vehicles type.

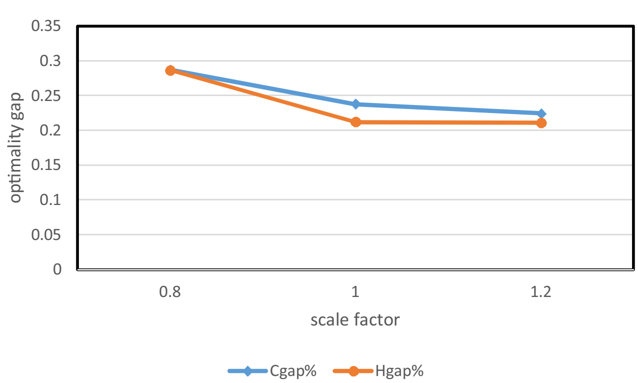
Scenarios 1–3 represents vehicles type from biggest capacity by the most expensive fixed cost to the smallest capacity and the cheapest fixed cost.



(a) Test instances number 1, 2 and 3



(b) Test instances number 4, 5 and 6



(c) Test instances number 7, 8 and 9

Fig. 9. CPLEX and IRDRP optimality gaps for different scale factors.

5. Conclusions

In this paper, we propose a heuristic algorithm for an IRDRP. Regional and time-based pricing are two common tactics for maximizing the revenue of firm. We incorporate these tactics for modeling dependency of customer demand on product offering prices. An effective heuristic algorithm is provided for an existing problem. This algorithm has different phases, including initialization, demand generation, demand adjustment, inventory-routing, and incorporating local search operators. Initialization constructs a solution representation structure. In the demand generation phase, a price optimization algorithm is incorporated for determining customers' offering prices during each period. If demand at each customer is not relatively small compared to vehicle capacity, a demand adjustment process is applied to adjust customer demand. We use a heuristic algorithm proposed by Abdelmaguid et al. (2009), in order to solve this IRP. Finally, local search operation explores the solution space by moving from one solution to its neighbor.

The experimental results, based on 35 instances of variable sizes, demonstrate the efficiency of our proposed algorithm. The results indicate that, as problem sizes increase, the IRDRP heuristic outperforms CPLEX. Sensitivity analysis is conducted to analyze the impact of available vehicle capacity and types on algorithm performance. According to the results, as the intensity of the scarce capacity setting increases, approaching an optimal solution becomes more difficult.

Further research could incorporate other price-response functions to evaluate algorithm performance under these functions. Various neighborhood search mechanisms can be attempted, to explore search space more efficiently. Individual consumer behavior could be considered in the context of demand modeling for predicting consumer reaction to product prices.

References

Abdelmaguid, T.F., Dessouky, M.M., 2006. A genetic algorithm approach to the integrated inventory-distribution problem. Int. J. Prod. Res. 44, 4445–4464.
Abdelmaguid, T.F., Dessouky, M.M., Ordonez, F., 2009. Heuristic approach for the inventory-routing problem with backlogging. Comput. Ind. Eng. 56, 1519–1534.
Anderson, H., Hoff, A., Christiansen, M., Hasle, G., Lokkhetangen, A., 2010. Industrial aspects and literature survey: combined inventory management and routing. Comput. Oper. 37, 1515–1536.
Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transp. Sci. 41 (3), 382–391.
Baita, F., Ukovich, W., Pesenti, R., Favaretto, D., 1998. Dynamic routing-and-inventory problems: a reiview. Transp. Res. A: Policy Pract. 8, 585–598.
Bell, W.J., Dalberto, L.M., Fisher, M.L., greenfield, A.J., Jaikumar, R., Kedia, P., Mack, R.G., Prutzman, P.J., 1983. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. Interfaces 13, 4–23.
Campbell, A.M., Savelsbergh, M.W.P., 2004. A decomposition approach for the inventory-routing problem. Transport. Sci. 38, 488–502.
Coelho, L.C., Laporte, G., 2013. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. Int. J. Prod. Res. 51 (23–24), 7156–7169.
Cordeau, J.-F., Lagana, D., Musmanno, R., Francesca, F., 2015. A decomposition-based heuristic for the multiple-product inventory-routing problem. Comput. Oper. Res. 55, 153–166.
Dror, M., Levy, L., 1986. A vehicle routing improvement algorithm comparision of a greedy and a matching implementation for inventory routing. Comput. Oper. Res. 13, 33–45.
Federgrun, A., Zipkin, P., 1984. Combined vehicle routing and inventory allocation problems. Oper. Res. 32, 1019–1037.
Leandro, C.C., Gilbert, L., 2013. The exact solution of several classes of inventory-routing problems. Comput. Oper. Res. 40, 558–565.
Leandro, C.C., Gilbert, L., 2014. Improved solutions for inventory-routing problems through valid inequalities and input ordering. Int. J. Prod. Econ. 155, 391–397.
Lenstra, J.K., Rinooy, K.A.H.G., 1981. Complexity of vehicle routing and scheduling problems. Networks 11, 221–227.
Liu, S.-C., Chen, J.-R., 2011. A heuristic method for the inventory routing and pricing problem in a suppy chain. Expert Syst. Appl. 38, 1447–1456.
Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equations of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092.
Mjirda, A., Jorbouni, B., Macedo, R., Hanafi, S., Mladenovic, N., 2014. A two phase variable neighborhood search for the multi-product inventory routing problem. Comput. Oper. Res. 52, 291–299.
Moin, N.H., Salhi, S., 2007. Inventory routing problems: a logistical overview. J. Oper. Res. Soc. 58, 1185–1194.
Moin, N.H., Salhi, S., Aziz, N.A.B., 2011. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. Int. J. Prod. Econ. 133 (1), 334–343.
Philips, R.L., 2005. Pricing and Revenue Optimization. University Press, California, Stanford.
Sarmiento, A.M., Nagi, R., 1999. A reveiw of integrated analysis of production-distribution systms. IIE Trans. 11, 1061–1074.

Shao, Y., Furman, K.C., Geol, V., Hoda, S., 2015. A hybrid heuristic strategy for liquified natural gas inventory routing. Transp. Res. C 53, 151–171.
Shen, Q., Chu, F., Chen, H., 2011. A Lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. Comput. Chem. Eng. 35, 2113–2123.
Shukla, N., Tiwari, M.K., Ceglarek, D., 2013. Genetic-algorithm-based algorithm portfolio for inventory routing problem with stochastic demand. Int. J. Prod. Res. 51 (1), 118–137.
Singh, T., Arbogast, J.E., Neagu, N., 2015. An incremental approach using local-search heuristic for inventory routing problem in industrial gases. Comput. Chem. Eng. 80, 199–210.
Talluri, K.T., Van Ryzin, G., 2004. The Theory and Practice of Revenue Management. Kluwer Academic Publishers, New York.
Vidovic, M., Popovic, D., Ratkovic, B., 2014. Mixed integer and heuristics model for the inventory routing problem in fuel delivery. Int. J. Prod. Econ. 147, 593–604.