An adaptive genetic algorithm for the time dependent inventory routing problem

Article *in* Journal of Intelligent Manufacturing · October 2014

DOI: 10.1007/s10845-012-0727-5

4 authors, including:

Young Hae Lee
Hanyang University

Mitsuo Gen
Fuzzy Logic Systems Institute

J Intell Manuf
DOI 10.1007/s10845-012-0727-5

An adaptive genetic algorithm for the time dependent inventory routing problem

Dong Won Cho · Young Hae Lee · Tae Youn Lee · Mitsuo Gen

Received: 15 June 2012 / Accepted: 26 December 2012

Abstract In this paper we propose an adaptive genetic algorithm that produces good quality solutions to the time dependent inventory routing problem (TDIRP) in which inventory control and time dependent vehicle routing decisions for a set of retailers are made simultaneously over a specific planning horizon. This work is motivated by the effect of dynamic traffic conditions in an urban context and the resulting inventory and transportation costs. We provide a mixed integer programming formulation for TDIRP. Since finding the optimal solutions for TDIRP is a NP-hard problem, an adaptive genetic algorithm is applied. We develop new genetic representation and design suitable crossover and mutation operators for the improvement phase. We use adaptive genetic operator proposed by Yun and Gen (Fuzzy Optim Decis Mak 2(2):161–175, 2003) for the automatic setting of the genetic parameter values. The comparison of results shows the significance of the designed AGA and demonstrates the capability of reaching solutions within 0.5 % of the optimum on sets of test problems.

Keywords Time dependent inventory routing problem · Adaptive genetic algorithm · Mixed integer programming

D. W. Cho · Y. H. Lee (✉) · T. Y. Lee
Department of Industrial and Management Engineering, Hanyang University, Ansan, Gyeonggi-do 426-791, South Korea
e-mail: yhlee@hanyang.ac.kr

D. W. Cho
e-mail: dwcmjcho@hanyang.ac.kr

T. Y. Lee
e-mail: taeyoun795@gmail.com

M. Gen
Fuzzy Logic Systems Institute, Iizuka, Japan
e-mail: mitsuogen@gmail.com

List of symbols

N	Number of retailers including depot
T	Number of time periods
K	Number of available vehicles during each period
M	Number of time intervals considered for each link
t	The starting time from the depot node 1
b_k	Volume capacity of vehicle k
C_i	Retailer's own capacity to hold inventory
d_{it}	Amount of demand at node i during period t
c_{ij}^{tm}	Travel time from node i to j if starting at i during time interval m at period t; $c_{ii}^{tm} = \infty$ for all i, m
s_{it}	Service time at node i at period t
T_{ij}^{tm}	Upper bound for time interval m for link (i, j) at period t
h_i^+	Holding cost per unit at node i
h_i^-	Backorder cost per unit at node i
c	Variable routing cost per hour
f_t	Fixed cost per vehicle at period t
B	$\mathrm{Max}_k b_k =$ capacity of largest vehicle

Decision variables

$$x_{ij}^{tm} \begin{cases} 1 & \text{if any vehicle travels directly from node } i \text{ to node } j \text{ starting from } i \text{ during time interval } m \text{ at period } t \\ 0 & \text{otherwise} \end{cases}$$

y_{ij}^{tm}	Amount transported on that trip for time interval m for link (i, j) at period t
I_{it}	Inventory levels at node i at period t
B_{it}	Inventory stock-outs at node i at period t
t_{it}	Departure time of any vehicle from node i at period t

Published online: 11 January 2013

Introduction

Traditionally, inventory control and vehicle routing problems have been studied separately (Abdelmaguid and Dessouky 2006). However, their integration, which is known as the inventory routing problem (IRP), has recently drawn more attention in the research and practice communities as a number of sophisticated supply chain management initiatives, such as vendor managed inventory (VMI) and continuous replenishment programs (CRP), have obtained widespread acceptance in many supply chain environments.

One of the current issues in IRP is the high degree of dynamics and uncertainty of traffic situations that influences many distribution systems. For example, traffic congestion causes a significant variation in travel speeds during rush hours. This is problematic for the typical IRP that do not explicitly consider changes and uncertainty in vehicle speeds due to traffic conditions. The main aim of IRP minimizes inventory and transportation costs while fulfilling customer demands (Baita et al. 1998). Therefore, inventory routing decisions that ignore these substantial speed variations result in inefficient, unrealistic, and suboptimal solutions. These poor decisions that guide freight vehicles into congested city traffic have a bad impact on overall supply chain performance (Kuo 2010). Moreover, customers have to wait for unreasonably long periods with unreliable information about distribution schedules. Under these circumstances, it becomes difficult to achieve the goal of IRP. This increases supply chain cost.

To aid in resolving this problem, we propose TDIRP in which inventory control and time dependent vehicle routing decisions are made simultaneously to meet customer demands for each time period over a specific planning horizon. In other words, TDIRP combines an inventory control problem with a time dependent vehicle routing problem (TDVRP). In the TDVRP, the travel time between two nodes of the network depends on the distance between the nodes and the time-dependent speed between these nodes (Malandraki and Daskin 1992). Therefore, TDIRP is a variation of IRP in which vehicle speeds between nodes do not rely on a constant value. On the other hand, TDIRP is a variation of TDVRP with the consideration of inventory control. In the past, the resolution of TDVRP is based on the minimal total route time of all vehicles without considering the inventory cost (Malandraki and Daskin 1992; Potvin et al. 2006; Balseiro et al. 2011). However, when only one element is evaluated and minimized, other elements are increased (Moin and Salhi 2007). Therefore, the time dependent vehicle routing and the inventory control decisions need to be determined concurrently to find the minimal supply chain cost.

Most IRP-related research assumes that the travel speeds (or times) between relevant locations (suppliers and customers) are constant (Baita et al. 1998; Liu and Chung 2009). However, in real life, the traffic conditions vary throughout the day (Kuo 2010). For example, during the rush hours in urban areas, the travel speeds are reduced substantially. In these environments, it becomes difficult to satisfy the time windows during which the customers must be served. This study is motivated by the effect of dynamic changes in traffic when time dependent vehicle routing is integrated with inventory control.

Examples of industries where TDIRP is applied or used are the grocery industry (supermarkets), the soft drink industry (vending machines), and the petrochemical industry (gas stations). In many distribution systems of these industries, suppliers determines when and how much to deliver to their customers over a finite planning horizon (Campbell et al. 2002). There are no customer orders. Instead, suppliers operate under the restriction that their customers are not allowed to run out of product (Campbell et al. 2002). In TDIRP, each day suppliers make decisions about when and which customers to visit and how much to deliver to each of them considering traffic conditions, while bearing in mind that decisions made today have an influence to some tasks which have to be done in the future. Therefore, some suppliers have a strategy to transport the goods to retail stores late at night or early in the morning to avoid congested traffic (Kuo 2010). But not all retail stores are open for 24 h. Due to the operation costs, there are many retail stores that only open when the number of customers is potentially high (Kuo 2010). During opening hours, the travel speeds are always lower and variable.

The objective of TDIRP is to find exactly when to deliver to a customer, how much to deliver to a customer in a time period and how to route vehicles with different travel speeds between the nodes such that the sum of transportation and inventory costs is minimized while still meeting customer demands. The IRP, which has lower complexity than the TDIRP, is known as a NP-hard problem due to the nature of its vehicle routing problem (Bell et al. 1983; Dror and Ball 1987; Campbell et al. 2002). Therefore, even solution approaches of simple IRPs are typically heuristics (Bell et al. 1983; Dror and Ball 1987; Bertazzi et al. 2002; Campbell et al. 2002; Abdelmaguid and Dessouky 2006).

Since the TDIRP is a generalization of the IRP, it belongs to the class of NP-hard problems. Moreover, TDVRP, which adds time-dependent travel times to VRP without the consideration of inventory control, is a NP-hard problem (Hill and Benton 1992; Ichoua et al. 2003; Malandraki and Daskin 1992). As a result, finding exact solutions to the TDIRP still eludes researchers, despite the impact of recent developments in mathematical programming and computer technology.

This paper addresses modeling issues, solution approaches and potential benefits for TDIRP. The main contributions are in two areas:

1. Most of the previous research has considered IRP that relies on a constant value to represent vehicle speeds. This study, to the best of our knowledge, is the first for TDIRP that includes time dependent vehicle speeds. TDIRP is formulated as a mixed integer programming (MIP) model. We find the optimal solution for TDIRP by using GAMS with Xpress solver.
2. Since finding the optimal solutions for TDIRP is a NP-hard problem, an adaptive genetic algorithm (AGA) is applied. We develop new chromosome structure to represent vehicle routing and inventory solutions in TDIRP. Then suitable crossover and mutation operators for the improvement phase are proposed. The computational results show that the AGA can produce solutions within 0.5 % of the optimum on sets of test problems within reasonable computation times.

This paper is organized as follows. Section two reviews relevant literature. The formal problem statement is described in section three. In section Four, the proposed AGA representation for the TDIRP is illustrated. The random generation procedure of the initial population and the designs of the crossover and mutation operators are presented. The experimentation and results are provided in section five, followed by the conclusions in section six.

Literature review

In this paper, we address a TDIRP that determines the integrated inventory and time dependent vehicle routing decisions over a multiple planning period under a situation in which backorders are permitted. Therefore, we focus on reviewing multi-period IRP and TDVRP. Hill and Benton (1992) proposed a parsimonious model for estimating the average travel speed in TDVRP in which a simple greedy heuristic is applied as a solution approach. Malandraki and Daskin (1992) represented mixed integer linear programming formulations of the TDVRP that treat the travel time functions as step functions. Several simple heuristic algorithms based on a nearest-neighbor heuristic are given for solving TDVRP. Ichoua et al. (2003) presented a TDVRP with soft time windows based on time-dependent travel speeds which satisfies the “first-in–first-out” property. They also proposed a parallel tabu search for TDVRP with soft time windows that minimizes the weighted sum of the total time and lateness. Chen et al. (2006) proposed a heuristic that comprises route construction and improvement to solve the real-time TDVRP with time windows. Donati et al. (2008) proposed a multi ant colony algorithm for solving TDVRP in a fleet of vehicles of fixed capacity. Kuo et al. (2009) proposed a model for assigning goods to vehicles and the vehicle routing problem with time-dependent travel speeds. They also proposed a tabu search algorithm for solving TDVRP. Balseiro et al. (2011) presented an ant colony algorithm hybridized with insertion heuristics for the TDVRP with Time Windows. However, TDVRP do not include inventory control which links time dependent vehicle routing decisions.

In the multi-period IRP, the objective is to find vehicle routes and quantities of goods shipped to retailers in each time such that the sum of transportation and inventory costs is minimized. Bertazzi et al. (2002) proposed a multi-period IRP and developed a solution heuristic based on an order up to level inventory policy in which whenever a customer is visited, its inventory will be refilled to the maximum level. Their research focused on problems of a single product and a single vehicle. Campbell et al. (2002) represented an optimization-based approach for the IRP and studied its practical application. In solving the IRP, they developed a two phase method. In the first phase, an integer programming model is provided to solve the inventory allocation part of the problem. In the second phase, vehicle routing and scheduling is solved using the delivery quantities decided in the first phase only as suggested values. Abdelmaguid and Dessouky (2006) proposed a multi-period IRP and formulated it as nonlinear programming with transportation costs represented as a nonlinear function. They developed a genetic algorithm (GA) for the multi-period IRP in which the genetic component was only used to find quantities of goods shipped to retailers in a time period. Therefore, in the proposed genetic representation, they concentrate on the delivery schedule and leave the vehicle routing part to be solved using any polynomial time heuristic. Abdelmaguid et al. (2009) also studied a multi-period IRP in which inventory holding, backlogging, and vehicle routing decisions are to be taken for a set of retailers who receive units of a single item from a depot with infinite supply. They developed constructive and improvement heuristics to obtain an approximate solution for this NP-hard problem and demonstrated its effectiveness through computational experiments. Moin et al. (2011) studied a many-to-one distribution network consisting of a depot, an assembly plant and many distinct suppliers where each provides a distinct product to the assembly plant. They consider a finite horizon, multi-periods, multi-suppliers, single warehouse and multi-products where a fleet of capacitated homogeneous vehicles, housed at a depot, transport products from the suppliers to meet the demand specified by the assembly plant in each period. They proposed a hybrid genetic algorithm which is based on the allocation first route second strategy. With the increase of problem size for the problems, the performance of the GA based algorithms increases and the results are obtained with significantly less computational times. Although the practical successes of IRP have been stressed in the literature, research papers on the time dependent vehicle routing problem with the consideration of inventory control are scant.

One key issue for our solution approach is to find the inventory and vehicle routing solutions that minimize the total cost of the TDIRP studied. However, as this problem is NP-hard, it is impossible to develop an exact algorithm that can solve the problem optimality in a reasonable time. GA has been successfully applied to a wide range of combinatorial optimization problems (Gen and Cheng 1997). The well-known applications include scheduling and sequencing, group technology, and many others (Chan et al. 2005, 2009; Chung et al. 2009; Kamrani and Gonzalez 2003; Onwubolu and Mutingi 2003; Tang and Liu 2002; Yang et al. 2012; Zandieh et al. 2010).

Moreover, various GA approaches have been developed to obtain good solutions for multi-period IRP. (Abdelmaguid and Dessouky 2006; Moin et al. 2011). The performance of GA relies highly on the setting values of the genetic parameters (Moon et al. 2006). Various studies have been performed to identify these correct settings for these values (Chan et al. 2005; Fogel et al. 2001; Moon et al. 2006; Yun 2002). Among these studies, the AGA approach is successfully implemented to various combinatorial optimization problems (Chan et al. 2005; Moon et al. 2006; Yun 2002).

Problem statement

This section presents a modeling framework for formulating TDIRP. The TDIRP is formulated as a mixed integer programming (MIP) model. The problem consists of a central depot with unlimited supply capacities and multiple retailers with deterministic demand to the case of single product over a multi-period. Therefore, a distribution system consists of a depot (or supplier), denoted 0, and geographically dispersed retailers, indexed $1, \ldots, N$. We have a directed graph $G(V, E)$ where $V = \{0, 1, \ldots, N\}$ is the finite set of nodes and E is the finite of directed links. A directed link is represented by an ordered pair of nodes (i, j) in which i is called the origin and j is called the destination of the link. A network is assumed complete and an $n \times n$ time dependent matrix $C(t) = \left[c_{ij}(t_i)\right]$ is also given representing the travel times on link $(i, j) \in E$, where $c_{ij}(t_i)$ is a known step function of the time of day, t_i, at the origin node i of the link. In this way, the day is divided into time intervals. Like Malandraki and Daskin (1992), once the time interval during which the vehicle starts traversing link (i, j) is known, the travel time of traversing link (i, j) is a known constant. Therefore, each link (i, j), from node i to node j, is replace by M_{ij} parallel links from node i to node j where M_{ij} is the number of distinct time intervals considered in the step function $c_{ij}(t_i)$ representing the travel time for the link. For clarity of exposition in the following formulation we simply denote the number of time intervals by M instead of M_{ij} as if the number of time intervals is the same for all physical links

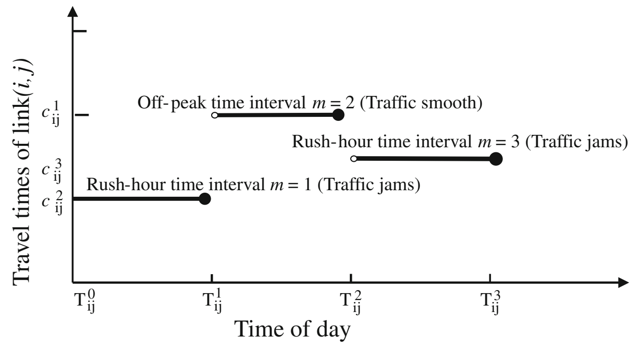

Fig. 1 Travel time step function for link (i, j) with three time intervals

in the network. A travel time step function for link (i, j) with three time intervals is shown in Fig. 1. There are three time intervals: the travel time of the first time interval $(m = 1)$ is c^1_{ij}, the travel time of the second time interval $(m = 2)$ is c^2_{ij} and the travel time of the last time interval $(m = 3)$ is c^3_{ij}.

The formulation concerns a TDIRP with exactly K capacitated homogeneous vehicles that permit waiting at the retailer nodes. It considers pick-ups (deliveries) from (to) all the retailers. The staring time from the supplier (depot) is given. Each retailer is served by one visit of one vehicle. We assume the following:

1. The travel time from node i to node j during time interval m is independent of the type of vehicle.
2. The pick-up (or delivery) time for each vehicle is independent of the type of vehicle and depends only on the retailer.
3. The depot has a sufficient supply of items that can cover all the retailer's demands throughout the planning horizon.
4. Vehicles must return to the depot at the end of the period, and no further delivery assignments should be made in the same period. In this model, we consider the case in which renting additional vehicles during the short planning horizon is not an option, and it is assumed that the fleet of vehicles remains unchanged throughout the planning horizon.

Based on these assumptions, we can formulate the TDIRP without considering which vehicle visits a node. Like Malandraki and Daskin (1992), the supplier (depot) node is also expanded as follows. Consider node 1 as the starting depot; that is, delete all the inbound links to node 1. Augment the network by K nodes $(n + 1, \ldots, n + K)$. These nodes correspond to the returning depot nodes for each of the K vehicles. Delete all outbound links from nodes $n+1, \ldots, n+K$ and all the links interconnecting the depot nodes. Thus, all vehicles start from the depot node 1 and each vehicle has to return to its specified return depot. Also set $c_{ij}(t_i) = c_{i1}(t_i)$ for every node $i = 2, \ldots, n$ and $j = n+1, \ldots, n+K$.

The notation used in our formulation is as follows. We will index the vehicles by k $(1, \ldots, K)$, the nodes including depot by i, j, h, l $(0, \ldots, N+K)$, the time intervals considered for each link by m $(1, \ldots, M)$ and time periods over a planning horizon by t $(1, \ldots, T)$.

With this notation the TDIRP is formulated as follows:

$$\text{Min} \sum_{t=1}^{T} \left[\sum_{j=1}^{N} \sum_{m=1}^{M} f_t x_{0j}^{tm} + \sum_{i=N+1}^{N+K} ct_{it} + \sum_{i=1}^{N} \left(h_i^+ I_{it} + h_i^- B_{it} \right) \right] \tag{1}$$

Subject to:

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} \sum_{m=1}^{M} x_{ij}^{tm} = 1, \quad j = 1, \ldots, N+K; \quad t = 1, \ldots, T \tag{2}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N+K} \sum_{m=1}^{M} x_{ij}^{tm} = 1, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T \tag{3}$$

$$\sum_{j=1}^{N} \sum_{m=1}^{M} x_{0j}^{tm} = K, \quad t = 1, \ldots, T \tag{4}$$

$$t_{0t} = t, \quad t = 1, \ldots, T \tag{5}$$

$$t_{jt} - t_{it} - B x_{ij}^{tm} \geq c_{ij}^{tm} + s_{jt} - B, \quad i = 0, \ldots, N; \quad j = 1, \ldots, N+K; \quad i \neq j; \quad t = 1, \ldots, T; \quad m = 1, \ldots, M \tag{6}$$

$$t + B x_{ij}^{tm} \leq T_{ij}^{tm} + B, \quad i = 0, \ldots, N; \quad j = 1, \ldots, N+K; \quad i \neq j; \quad t = 1, \ldots, T; \quad m = 1, \ldots, M \tag{7}$$

$$t - T_{ij}^{tm-1} x_{ij}^{tm} \geq 0, \quad i = 0, \ldots, N; \quad j = 1, \ldots, N+K; \quad i \neq j; \quad t = 1, \ldots, T; \quad m = 1, \ldots, M \tag{8}$$

$$y_{ij}^{tm} - b_k x_{ij}^{tm} \leq 0, \quad i = 0, \ldots, N; \quad j = 1, \ldots, N+K; \quad i \neq j; \quad t = 1, \ldots, T; \quad m = 1, \ldots, M \tag{9}$$

$$\sum_{\substack{h=1 \\ h \neq i}}^{N+K} \sum_{m=1}^{M} y_{ih}^{tm} - \sum_{\substack{l=0 \\ l \neq i}}^{N} \sum_{m=1}^{M} y_{li}^{tm} \leq 0, \quad i = 0, \ldots, N; \quad t = 1, \ldots, T \tag{10}$$

$$I_{it-1} - B_{it-1} - I_{it} + B_{it} + \left(\sum_{\substack{h=0 \\ h \neq i}}^{N} \sum_{m=1}^{M} y_{hi}^{tm} - \sum_{\substack{l=1 \\ l \neq i}}^{N+K} \sum_{m=1}^{M} y_{il}^{tm} \right) = d_{it}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T \tag{11}$$

$$I_{it} \leq C_i, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T \tag{12}$$

$$I_{it} \geq 0, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T \tag{13}$$

$$B_{it} \geq 0, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T \tag{14}$$

$$x_{ij}^{tm} = 0 \text{ or } 1, \quad i = 0, \ldots, N; \quad j = 1, \ldots, N+K; \quad i \neq j; \quad t = 1, \ldots, T; \quad m = 1, \ldots, M \tag{15}$$

$$t_{it} \geq 0, \quad i = 0, \ldots, N; \quad t = 1, \ldots, T; \tag{16}$$

$$y_{ij}^{tm} \geq 0, \quad i = 0, \ldots, N; \quad j = 1, \ldots, N+K; \quad i \neq j; \quad t = 1, \ldots, T; \quad m = 1, \ldots, M \tag{17}$$

$$I_{0t} = 0, \quad t = 1, \ldots, T \tag{18}$$

$$B_{0t} = 0, \quad t = 1, \ldots, T \tag{19}$$

The objective function (1) includes transportation costs based on fixed usage and the total route time (travel time plus service time plus waiting time) and inventory carrying and shortage costs on the end of period inventory positions. Constraints (2) to (3) ensure that a vehicle will visit a location no more than once in a time period. Constraint (4) makes sure that exactly K vehicles are used in a time period. Constraint (5) sets the starting time from depot node 1 equal to t for all vehicles. Constraint (6) computes the departure time at node j. Constraints (7) and (8) make sure that the proper parallel link m is chosen between nodes i and j according to the departure time from node i. Constraint (9) serves two purposes. The first one is to make sure that the amount transported between two locations will always be zero whenever there is no vehicle moving between these locations, and the second is to ensure that the amount transported is less than or equal to the vehicle's capacity. Constraint (10) is necessary to eliminate sub-tours. Constraint (11) is the inventory balance equations for the retailers. Constraint (12) limits the inventory level of the retailers to the corresponding storage capacity. It is assumed that the amount consumed by each retailer in a given period is not kept in the retailer's storage location; accordingly, it is not accounted for in constraint (12). Constraints (13)–(19) are the domain constraints.

Adaptive genetic algorithm for TDIRP

GA is a stochastic search optimization method based on the mechanism of natural selection and natural genetics

Table 1 Cost and service time information for the small sample problem

Costs and time	Retailer		
	1	2	3
Unit holding cost ($ per unit per period)	0.11	0.12	0.08
Unit shortage cost ($ per unit per period)	3.45	2.79	2.38
Service time	8.00	8.00	8.00

Table 2 Travel speed matrix for the sample problem

Link	Categories	Speed levels	
		Rush-hour time speed Time interval 1 (0–50)	Off-peak time speed Time interval 2 (50–100)
(i, j)	1	0.54	0.81
	2	0.81	1.22
	3	1.22	1.82

(Gen and Cheng 1997). This optimization process starts by generating a population of random solutions (individuals), which are in the format of a genotype. The specification of a solution in the population can be stored in one or more chromosomes where a chromosome, which represents a solution to the problem at hand, is by itself made of an ordered sequence of single genes. The chromosomes evolve through successive iterations, called generations. Once the evaluation of chromosomes in each generation is carried out using some measure of fitness, parents are selected and a crossover mechanism is applied to obtain a new generation of individuals (offspring). Moreover, a mutation scheme is also applied to modify a chromosome in the population. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum or suboptimal solution to the problem (Gen and Cheng 1997). As mentioned earlier, we introduce an AGA approach to identify the correct settings of the genetic parameter values (such as population size, crossover and mutation rates) for TDIRP. In the following subsections we report a detailed description about the developed AGA.

The proposed genetic representation

Prior to the application of GA to TDIRP, it is important to define how to encode a generic solution of the problem into a chromosome suitable to the application of genetic operators. Designing a suitable chromosomal representation is one of the important issues that affect the successful performance of GA (Gen and Cheng 1997).

To illustrate the proposed chromosome representation, consider a small sample problem in which the distribution system consists of single supplier, three retailers with a storage capacity of 100 units each, inventory holding and shortage costs, and service time as shown in Table 1.

At the beginning of the planning period, all retailers have zero inventory positions. Two vehicles, each with 75 units of capacity, are available to serve the retailers in every period for a 3-period planning horizon. In the example, the travel speed matrix is reported in Table 2. For each link, there are 3 categories of time-dependent speed as shown in Table 2.

There are two speed levels: rush-hour time speed and off-peak time speed. Rush-hour time speed is assumed for transport during time interval 1 called rush hours and off-peak time speed is assumed for transport during time interval 2 called off-peak hours. For example, for journeys in Category 1 the travel speed is 0.54 during time interval 1 and 0.81 during time interval 2. When a vehicle is decided to be used in any period, $10 is charged. The transportation cost per unit time is set to $1. The distribution network and distance between nodes are described in the graph shown in Fig. 1. Thus, transportation time between nodes is calculated by dividing the distance by transport velocity. As a result, transportation cost is calculated by multiplying transportation time by transportation cost per unit time. For example, if a transport travels link (1, 2) for journeys in Category 1 during time interval 1, transportation cost is 22.22(= (12 \times \$1)/0.54)$ because the speed is 0.54 from Table 2. The demand requirements for every period in the planning horizon are also provided in Fig. 2.

Figure 3 illustrates an example of the chromosome structure for a sample solution with the demand matrix given in Fig. 2.

We define a chromosome which combines the delivery schedule part with the vehicle routing part as shown in Fig. 3. The genetic representation of a sample solution takes the form of a two dimensional matrix with three rows and twelve columns. Each row in the matrix corresponds to the planning periods from 1 to 3. And as the delivery schedule part, the column 1, 2 and 3 is the index of retailers in which each cell in the matrix contains the amount to be delivered to the corresponding retailer (given in the column) and the corresponding period (given in the row). After the index of retailers, the next columns represent the vehicle routing part. The column 4, 5 and 6 is the vehicle route sequence with the index of vehicles referred to vehicle 1 in the column 7 in which each cell is the number of visited retailers by vehicle 1 for the corresponding route sequence (given in the column) and the corresponding period (given in the row). After these columns, the column structure for each vehicle in the vehicle routing part of the matrix is the same as that of the column 4, 5, 6 and 7. It can be seen that the route of vehicle 1 is defined by depot—retailer 1—retailer 2— depot in period 1 where a collection of 34 and

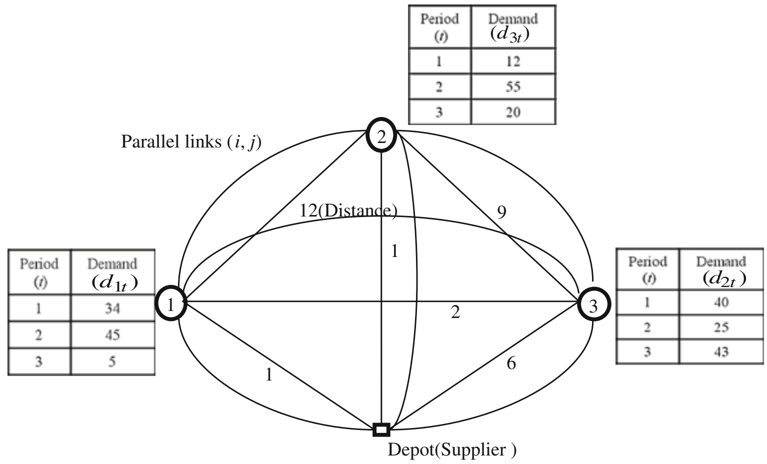

Fig. 2 Distribution network and demand requirements for the sample problem

Fig. 3 Genetic representation for a sample solution

		Delivery schedule			Vehicle routing							
		Retailer			Route sequence			Vehicle	Route sequence			Vehicle
		1	2	3	1	2	3		1	2	3	
Period	1	34	12	40	1	2	0	1	3	0	0	2
	2	45	55	25	1	3	0	1	2	0	0	2
	3	5	20	43	1	2	0	1	3	0	0	2

12 units is delivered. On the one hand, the route of vehicle 2 is defined by the depot—retailer 3—depot in period 1 where 40 units are delivered. Based on the scheduled delivery amounts in the matrix, the inventory position of each retailer can be easily determined. Note that the given solution structure is composed of integer type members.

Fitness evaluation and selection

The fitness value is the measure of goodness of a solution with respect to the original objective function. For the TDIRP we have considered, candidate solutions with lower total costs imply better solutions. Thus, the fitness function for each chromosome is defined as the inverse to its total cost value with the form in Eq. (20). In result, the higher the fitness value the more chances the individuals have to be selected.

$$\text{Fitness} = \frac{1}{\sum_{t=1}^{T}\left[\sum_{j=1}^{N}\sum_{m=1}^{M} f_t x_{0j}^{tm} + \sum_{i=N+1}^{N+K} ct_{it} + \sum_{i=1}^{N}\left(h_i^{+} I_{it} + h_i^{-} B_{it}\right)\right]} \tag{20}$$

In our GA approach, we use a roulette-wheel selector with a selection probability equal to fitness value for each solution, and a GA search structure with elite preservation is applied.

Initialization

The quality and the size of the initial population can have a significant impact on the efficiency of a GA. As a result, a carefully developed heuristic is required to generate the randomly initial populations within the constrained solution space. The population size is chosen by measuring the convergence time of the GA through a trial and error approach. Based on this observation, we propose a constructive heuristic, which is used to randomly generate new chromosomes that can satisfy most of the constraints. The GA initialization procedure is conducted as follows.

Initialization procedure

Step 1: Start with period index $t = 1$.
Step 2: Generate random value of delivery amount for each retailer i in period t.
Step 3: Divide retailers that need delivery in period t into K delivery groups randomly.
Step 4: Assign K delivery groups to K available vehicles in period t randomly.
Step 5: Generate vehicle routing using nearest neighbor search (NSS) heuristic for the delivery groups allocated to vehicles. The NSS heuristic may be describes as follows:

NNS 1: Start from the depot at the given starting time.
NNS 2: Find a nonvisited node such that the link from the last-visited node to this node has the smallest time of traversal and does not violate the route time requirements and the vehicle capacity restrictions for the current vehicle. Add this node to the tour of the current vehicle.
NNS 3: Repeat NNS 2 until all nodes are visited and return to the depot. If all nodes are visited, stop. Otherwise,

if the link does violate either the route time requirements or the capacity restrictions for the current vehicle, indicate infeasible solution and stop.

Step 5: Decide how much to be delivered in period t for each retailer by instantiating remaining vehicle capacity (RVC) algorithm for improving initial feasible solutions.
Step 6: Set $t = t + 1$.
Step 7: Return to step 2 until $t \leq T$.

It is common to randomly generate the initial populations in the GA initialization process. The NNS heuristic indicates infeasibility when the solutions violate hard constraints in our model. The RVC heuristic is only applied to the initially feasible solutions generated from an NNS heuristic to obtain a good initial population.

Let w_{it} be the amount of delivery to retailer i in period t. For retailers that need no delivery, their $w_{it} = 0$. Let DL_{kt} represent the total remaining vehicle capacity for vehicle k in period t after assigning the delivery amounts and determining the vehicle route in the solution of NNS heuristic. For the remaining capacity of vehicle K in period t, if any, we need to decide whether it is beneficial to allocate this capacity to meet future retailer delivery. Here, we only consider meeting future delivery for retailers allocated to vehicle K that have planned deliveries in period $t + 1$. Let MD_k be the set of retailers assigned to vehicle k with $w_{it} > 0$ among the retailers for vehicle k in period t. And let FD_k be the set of retailers assigned to vehicle k with $w_{it+1} > 0$ among the retailers for vehicle k in period t. We present here an RVC heuristic described in the following list.

RVC heuristic procedure

Step 1: For every retailer $i \in \left(MD_k \bigcap FD_k\right)$ calculate DL_{kt} of vehicle k in period t.
Step 2: Sort retailers in set $MD_k \bigcap FD_k$ in a routing order of vehicle k.
Step 3: For each retailer i in the ordered set $MD_k \bigcap BD_k$ do

If $DL_{kt} \geq w_{it+1}$, let $DL_{kt} = DL_{kt} - w_{it+1}$, $w_{it} = w_{it} + w_{it+1}$ and $w_{it+1} = 0$, remove i from $MD_k \bigcap FD_k$ in period t.
If $DL_{kt} < w_{it+1}$, let $DL_{kt} = 0$,$w_{it} = w_{it} + w_{it+1} - DL_{kt}$ and $w_{it} = 0$, remove i from $MD_k \bigcap BD_k$ in period t.

Step 4: If $\left(MD_k \bigcap FD_k\right) = \varphi$ or $DL_{kt} \leq 0$ then stop. Else go to Step 3.

In the case where deliveries made to the retailer involve less than vehicle-load quantities, it is profitable to place more than one delivery on the same vehicle for more efficient transport. The RVC heuristic determines how much to add to the deliveries made to the retailers in period t only for deliveries that they have planned on the same vehicle in period $t + 1$. Partial fulfillment of the delivery occurring at a retailer in period $t + 1$is considered. Thus, the RVC heuristic generates solutions in which delivery schedules partially or totally cover retailer delivery requirements in period $t+1$. For the set of retailers that has the amount of delivery in period $t+1$, the cost saving that can be achieved by adding a retailer delivery in period $t + 1$ to its current delivery is applied. The RVC heuristic could have been applied to deliveries that they have planned beyond period $t + 1$. However, for this particular problem, the RVC heuristic is found to be too time consuming and costly. The process of calculating the total cost of any chromosome is repeatedly used as a subroutine during the course of the RVC heuristic whenever a change in the chromosome occurs. Candidate solutions with lower total costs imply better solutions as described in section "Fitness evaluation and selection".

Crossover operator

Crossover is the main operators of GA, as it can inherit better features from the fittest solutions among the generations. It operates two chromosomes at a time and produces offspring by combining both chromosomes' characteristics. A traditional way to achieve crossover would be conducted by breaking the chromosome and exchanging the broken genes between two different parent solutions. This results in two new offspring solutions with combined characteristics from both parent solutions.

For the designed chromosome representation of the TDIRP, the two-dimensional matrix structure can be broken either vertically or horizontally or by a combination of them for delivery schedule part or vehicle routing part, respectively. A horizontal or vertical breakdown of the delivery schedule part means that delivery schedules will be exchanged between the two parent solutions. On the one hand, a horizontal or vertical breakdown of the vehicle routing part means that the vehicle tour will be exchanged between the two parent solutions. A horizontal breakdown of the delivery schedules for a selected set of periods will maintain the vehicle capacity constraints but may result in a retailer storage capacity violation. This is similar to the one used by Abdelmaguid and Dessouky (2006) for a different IRP. They address the problem that solutions generated by the horizontal breakdown of delivery schedules for a selected set of periods will generally have poor inventory decisions that appear in the form of extra unnecessary inventory or backorder. On the other hand, in our study, a vertical breakdown of vehicle routing for a selected set of vehicle route sequences may violate both the vehicle capacity restrictions

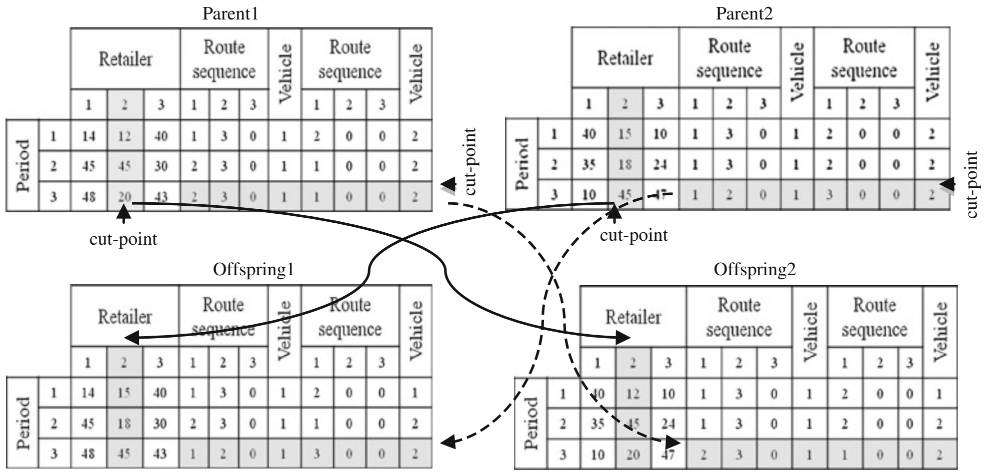

Fig. 4 Illustration of a vertical breakdown of delivery schedules and a horizontal breakdown of vehicle routing

and the route time requirements, which discouraged us from applying this method.

Figure 4 illustrates an example of a vertical breakdown of delivery schedules for a selected set of retailers and a horizontal breakdown of vehicle routing for a selected set of periods which will be exchanged between the two parent solutions. This breakdown may violate either the vehicle capacity restrictions or the route time requirements. If a repair algorithm to ensure these particular constraints is applied to infeasible solutions, solutions generated by this breakdown will generally have good inventory and transportation decisions that appear in the form of inventory, backorder and transportation cost, which encouraged us in applying this method.

Repair algorithms enjoy a particular popularity in the GA: for many combinatorial problems (e. g., traveling salesman problem, knapsack problem, set covering problem, etc.) it is relatively easy to repair an infeasible individual (Michalewicz 1996). To restore feasibility for violating either the vehicle capacity restrictions or the route time requirements, we design a special repair algorithm which would repair a chromosome by adjusting the delivery amounts or changing the vehicle routings, moving it back into the search space. The vehicle capacity fixation (VCF) algorithm and the route time fixation (RTF) algorithm, which fixes the violation of vehicle capacity restrictions and route time requirements, will be described in section "Fixation algorithm for violating vehicle capacity and route time restrictions".

Fixation algorithm for violating vehicle capacity and route time restrictions

Let $\delta_{it} = d_{it} - I_{it-1} + B_{it-1}$ be the delivery amounts of retailer i at the beginning of period t, and let ND_k be the set of retailers assigned to vehicle k that have $\delta_{it} > 0$. For retailers that have $\delta_{it} \leq 0$, no delivery is needed, and accordingly their $w_{it} = 0$. Let FD_k be the set of retailers which belong to vehicle route violating the vehicle capacity restrictions for vehicle k in period t.

VCF procedure

Step 1: For every retailer $i \in ND_k$ let $\Delta b_k = \sum_{i \in ND_k} \delta_{it} - b_k$.
Step 2: Sort retailers in set ND_k in a non-decreasing order.
Step 3: For each retailer i in the ordered set ND_k do

If $\Delta b_k \geq \delta_{it}$, let $\Delta b_k = \Delta b_k - \delta_{it}$, $w_{it} = 0$, remove i from ND_k.
If $\delta_{it} > \Delta b_k > 0$, let $w_{it} = \delta_{it} - \Delta b_k$ and $\Delta b_k = 0$.
If $\Delta b_k \leq 0$, let $w_{jt} = \delta_{jt}$ for all j in ND_k with unassigned w_{jt} values and stop.
Continue.

In the VCF algorithm, if the available vehicle capacity can cover all retailers demand values, the delivery amount w_{it} for each retailer is set to δ_{it}. Otherwise, delivery amounts are assigned to the retailers such that the partial delivery amount is assigned to the retailer with the lowest shortage amount among the retailers whose $\delta_{it} > b_k$ first.

Let CT_{kt} represent the completion time of the route for vehicle k in period t. Let RD_{kt} be the set of retailers which belong to vehicle route violating the route time requirements for vehicle k in period t. Let DRD_{kt} be the set of retailers deleted in set RD_{kt}. The following procedure describes the main steps of the RTF algorithm used to fix the route time requirement violation.

RTF procedure

Step 1: For every retailer $i \in RD_{kt}$, let $\Delta t = 1$ unit period and $\Delta g_k = CT_{kt} - \Delta t$.
Step 2: If $CT_{kt} \geq 2\Delta t$, indicate infeasible solution and stop.

Step 3: Sort retailers in a route order for the retailers, $i \in RD_{kt}$.

Step 4: For each retailer i in a route order of the set RD_{kt} do

If $\Delta g_k \geq x_{ij}^{tm}$, let $\Delta g_k = \Delta g_k - x_{ij}^{tm}$, $x_{ij}^{tm} = 0$ and $CT_{kt} = CT_{kt} - x_{ij}^{tm}$ and remove i from RD_{kt} and add i to DRD_{kt}.
If $x_{ij}^{tm} > \Delta g_k > 0$, let $x_{ij}^{tm} = 0$, $CT_{kt} = CT_{kt} - x_{ij}^{tm}$ and $\Delta g_k = 0$ and remove i from RD_{kt} and add i to DRD_{kt}.
If $\Delta g_k \leq 0$, estimate x_{ij}^{tm} for all i, j in RD_{kt} with unassigned x_{ij}^{tm} values and go to step 5.
Continue.

Step 5: Search for the set ND_{lt} for the retailers of the vehicle l with the lowest total route time among vehicles which do not violate the route time requirements in period t and replace the vehicle l for the vehicle $k$for the set DRD_{kt} for the retailers removed from RD_{kt} in period t.

Step 6: Generate the route of vehicle l by applying NN heuristics to each retailer i in the set $DRD_{lt} \bigcup ND_{lt}$ in period t and estimate CT_{lt} for its route time.

Step 7: If $CT_{lt} \geq \Delta t$, indicate infeasible solution and stop.

In the RTF algorithm, the vehicle with the lowest total route time among vehicles which does not violate the route time requirements is exploited.

Mutation operator

Mutation operators play a major role which generates spontaneous random changes in various chromosomes. In result, they help the GA to reach global solutions in the search space. They are applied to each offspring resulting from the crossover operator. The simple way of the mutation operator is to randomly mutate a solution's genes and hence generate a new solution that is not presented from the original one.

We observe that this type of representation tends to make interactions between delivery schedules and vehicle tours. We design a mutation operator that is suitable for this task. The algorithm attempts to alter the vehicle routing with retailer swap as well as the delivery schedule with delivery exchange for a randomly selected chromosome.

The process of transferring part of a retailer's delivery amount from one period to another is referred to as the delivery exchange. The algorithm of the delivery exchange attempts to transfer some amount of the product picked up in the future/past period to the current selected period. If the amount is to be transferred from a period to one of its successors, this delivery exchange is called forward delivery exchange. Similarly, if the amount is to be transferred to a preceding period, this operation is called backward delivery exchange.

Let VD_{kt} be the set of vehicles that have $DL_{kt} > 0$ in period t. And let BD_k be the set of retailers assigned to vehicle k with $w_{it-1} > 0$ among the retailers for vehicle k in period t. On the one hand, FD_k is a set of retailers assigned to vehicle k with $w_{it+1} > 0$ among the retailers for vehicle k in period t. We present here the algorithm of the delivery exchange described in the following list.

Delivery exchange procedure

Step 1: Select randomly a period t.

Step 2: For every vehicle $k \in VD_{kt}$, select the vehicle l with the highest remaining vehicle capacity in period t.

Step 3: If the selected period is the first period, go to step 6.

If the selected period is the last period, go to step 8.
Else go to step 4.

Step 4: Select one of forward and backward delivery change randomly.

Step 5: If forward delivery change is selected, go to step 6.

If backward delivery change is selected, go to step 8.

Step 6: Select the retailer i in the set $i \in FD_l$ randomly.

Step 7: If $DL_{lt} \geq w_{it+1}$, let $w_{it} = w_{it} + w_{it+1}$, $w_{it} = 0$ and stop.

If $DL_{lt} < w_{it+1}$, let $w_{it} = w_{it} + w_{it+1} - DL_{lt}$, $w_{it} = 0$ and stop.

Step 8: Select the retailer i in the set $i \in BD_l$ randomly.

Step 9: If $DL_{lt} \geq w_{it-1}$, let $w_{it} = w_{it} + w_{it-1}$, $w_{it} = 0$ and stop.

If $DL_{lt} < w_{it-1}$, let $w_{it} = w_{it} + w_{it-1} - DL_{lt}$, $w_{it} = 0$ and stop.

In the algorithm of delivery exchange, a vehicle is found with the highest remaining vehicle capacity in a given period. A randomly selected delivery exchange method is applied to the delivery schedule of each retailer assigned to the selected vehicle. Then, some amount of the product picked up in the future/past period will be transferred to the current selected period.

The process of swapping two retailers assigned to different vehicle routes is referred to as a retailer swap. The algorithm of a retailer swap swaps two retailers which are randomly and respectively selected from two vehicle routes in a given period. As presented earlier in the designed crossover operator, a solution alternative in which two vehicle routes reciprocally exchange retailers in a given period is not generated. Such a retailer swap helps the GA to reach further solutions

Period		Retailer 1	Retailer 2	Retailer 3	Route sequence 1	Route sequence 2	Route sequence 3	Vehicle	Route sequence 1	Route sequence 2	Route sequence 3	Vehicle	Vehicle1 Remaining vehicle capacity (75)	Vehicle2 Remaining vehicle capacity (75)	Vehicle1 Remaining route time capacity (100)	Vehicle2 Remaining route time capacity (100)
Period	1	34	15	43	1	2	0	1	3	0	0	2	26	32	35	67
	2	45	30	30	2	3	0	1	1	0	0	2	15	30	14	36
	3	35	20	43	2	3	0	1	1	0	0	2	12	40	23	50

Period		Retailer 1	Retailer 2	Retailer 3	Route sequence 1	Route sequence 2	Route sequence 3	Vehicle	Route sequence 1	Route sequence 2	Route sequence 3	Vehicle	Vehicle1 Remaining vehicle capacity (75)	Vehicle2 Remaining vehicle capacity (75)	Vehicle1 Remaining route time capacity (100)	Vehicle2 Remaining route time capacity (100)
Period	1	39	32	43	3	2	0	1	1	0	0	2	0	36	65	75
	2	0	33	42	2	3	0	1	0	0	0	2	0	75	14	100
	3	75	0	31	3	0	0	1	1	0	0	2	44	0	63	50

Fig. 5 Illustration of delivery exchange and a retailer swap

in the search space. The following procedure describes the main steps of the retailer swap algorithm.

Retailer swap procedure

Step 1: Select a period t randomly.
Step 2: Select two vehicles in a given period t randomly.
Step 3: Swap two retailers respectively and randomly selected among the retailers assigned to each vehicle.
Step 4: Generate vehicle routing by applying NNS heuristic to new delivery groups respectively.
Step 5: If the resultant children violate either the vehicle capacity restrictions or the route time requirements, instantiate and apply VCF or RTF algorithm.

The process of the retailer swap from one vehicle tour to another is simple. However, the resultant children may violate either the vehicle capacity restrictions or the route time requirements. Therefore, the VCF or RTF algorithm needs to be applied to restore feasibility. Figure 5 shows an example for how forward/backward the delivery exchange and retailer swap operates.

Adaptive operator

An AGA approach with an adaptive scheme could automatically modulate crossover and mutation rates. The objective of the adaptive scheme is to enhance the searching ability by increasing the local search and global search ability. Although GA has been successfully applied to IRPs (Abdelmaguid and Dessouky 2006; Moin et al. 2011), it has been only applied to the delivery schedule part of the IRPs. In this study, we develop new chromosome structure to apply GA to TDIRP. Then, suitable crossover and mutation operators for the improvement phase are developed. Since the correct setting values of the genetic parameters (such a population size, crossover and mutation rates) for the TDIRP is not an easy task, we use an adaptive genetic operator proposed by Yun and Gen (2003).

For adaptive schemes, they include the fitness values of the parent and offspring at each generation to regulate the adaptive crossover and mutation rates. This scheme involves some rules to adaptively adjust the crossover and mutation rates according to the performance of the genetic operators. It increases the probability of the occurrence of the crossover and mutation operator if it consistently produces better offspring during genetic search process; however, it also reduces the probability of the occurrence of the crossover and mutation operator if it always produces a poorer offspring. This scheme is based on the fact that it promotes the well-performing operators to produce more offspring while also weakening the chance for poorly performing operators to destroy the potential individuals during the genetic search process. The following procedure describes the main steps of the adaptive operator.
Procedure: regulation of crossover and mutation operators

Begin

– If $\left(\overline{f_{par\_size}}(t)\,/\,\overline{f_{off\_size}}(t)\right) - 1 \geq 0.1$ then

$$Pc\,(t+1) = Pc\,(t) + 0.05,$$
$$Pm\,(t+1) = Pm\,(t) + 0.005,$$

– If $\left(\overline{f_{par\_size}}(t)\,/\,\overline{f_{off\_size}}(t)\right) - 1 \geq 0.1$ then

$$Pc\,(t+1) = Pc\,(t) - 0.05,$$
$$Pm\,(t+1) = Pm\,(t) - 0.005,$$

- If $\left(-0.1 < \overline{f_{par\_size}}(t) / \overline{f_{off\_size}}(t)\right) - 1 < 0.1$ then

$$Pc(t+1) = Pc(t),\ Pm(t+1) = Pm(t),$$

End
Par_size: Parent size
Off_size: Offspring size
$\overline{f_{par\_size}}(t) / \overline{f_{off\_size}}(t)$: The average fitness value of parents and offesprings.
$Pc(t)\,\&Pm(t)$: *The* rate of crossover and mutation at generation t.

Numerical experiments and results

Adaptive genetic algorithm implementation

In our AGA implementation for TRIRP, we employ an AGA search process with elite preservation. The algorithm starts by generating the initial population using constructive heuristics. The population size remains constant throughout the application of the algorithm. The designed crossover and mutation operators are applied for a randomly selected pair of solutions from the current population. Then, the algorithm begins the improvement process of solutions to yield an improved offspring. We use a simple roulette-wheel selector with the selection probability for each solution inversely proportional to its total cost value. An adaptive genetic operator with an adaptive scheme could automatically modulate crossover and mutation rates. To replace the current population with a new one, the selection phase followed by the crossover and the mutation operators is repeated a number of times equal to the population size. The creation of a new population is repeated a number of times called the number of generations. A set of the best solutions found is stored in what is referred to as the elite set. This elite set has a fixed size and is used to provide the starting population of solutions in every generation. In our experimentation, we used the following parameters. Number of generations: 300; population size: 60; elite size: 20. Adaptive genetic operator heuristic automatically regulates crossover and mutation probability. The adjusted rates should not exceed a range from 0.5 to 0.8 for the rates of crossover and mutation operators.

Experimental design

The prototype of the proposed AGA algorithm is developed in a C++ programming platform and executed on a Pentium(R) Dual-core, 2.93 GHz PC with 1 GB of RAM. We compare the results obtained by the GA with the optimal solutions to give a measure of how far the results are from them. The optimal solutions are generated using GAMS with Xpress solver 2.0.36.7, a commercial MIP solver. Richard et al. (2009) show that some of the innovations in MIP solution technology that have been implemented in Xpress solver have resulted in vast improvements in solution times, so that problems that were previously considered to be intractable are now solved routinely within a short period of time. Atamturk and Savelsbergh (2005) review recent developments in integer-programming software systems: Cplex, Lindo, and Xpress-mp and discuss the options available to users for adjusting the behavior of these solvers when the default settings do not achieve the desired performance level.

Random test problems are generated. We assume that retailers are allocated in a square of 20×20 distance units, and their coordinates are generated using a uniform distribution within these limits. The depot is located in the middle of the square. The transportation cost per unit time is set to 1. The fixed usage cost per vehicle is set to 10. The retailer unit holding costs are generated using a normal distribution with a mean of 0.1 and a standard deviation of 0.02, and each retailer has a storage capacity of 120 units. The retailer unit shortage costs are generated using a normal distribution with a mean of 3 and a standard deviation of 0.5, and the retailer demands are generated using a uniform distribution from 5 to 50 units per day. At the beginning of the planning period, retailers have zero inventory and zero shortage positions. The number of homogenous vehicles is 2.

The travel speed matrices for the three scenarios given by Ichoua et al. (2003) are shown in Table 3.

The travel times were calculated using a 3×3 time-dependent travel speed matrix where each row corresponds to a category of link and each column to a time interval. Within the scheduling horizon, the first and third intervals stand for the rush hours. The second interval corresponds to the off hours, when the traffic density is lower. The entries of the travel speed matrix were adjusted to create three different types of scenarios. For each scenario, the travel speeds in the rush hours were obtained by dividing the travel speeds in the off hours by a factor α. In scenarios 1, 2 and 3, α is set to 1.5, 2 and 4, respectively. Hence, scenario 3 is the one with the highest degree of time-dependency, while scenario 1 is the one with the lowest. We randomly generate category and time interval for each scenario.

A total of 24 test problems were generated by varying the number of retailers (N) the number of planning periods (T), and three different degrees of time-dependency for travel speed. We generate four levels of N(5, 15, 30, and 45) and two levels of T (5 and 7). The total vehicle capacity is selected to be fixed at 150, 450, 900 and 1350 for each level of N, respectively.

The naming convention of the test problems starts with the letters "TDIRP", followed by two digits for the number of retailers. The third digit represents the length of the planning horizon. Finally, the scenario number is given at the last digit and is separated from the former digits by a hypen. Thus, the problem TDIRP055-1 represents the first randomly

Table 3 Travel speed matrices for experiment (Ichoua et al. 2003)

Scenarios	Categories	Speed levels		
		Time interval 1	Time interval 2	Time interval 3
		(0–60)	(60–120)	(120–180)
1	1	0.54	0.81	0.54
	2	0.81	1.22	0.81
	3	1.22	1.82	1.22
2	1	0.33	0.67	0.33
	2	0.67	1.33	0.67
	3	1.33	2.67	1.33
3	1	0.12	0.46	0.12
	2	0.46	1.92	0.46
	3	0.96	3.84	0.96

generated test problem with 5 retailers, a planning horizon of 5 periods and scenario 1.

We also analyze the impact of the number of retailers, the number of planning periods, and degree of time-dependency for the travel speed on the total cost and examine the interaction among the number of retailers, the number of planning periods, and degree of time-dependency for the travel speed. Therefore, three groups of independent factors are investigated through the experimental design. The number of levels of these factors with their respective values is listed in Table 4.

Results and discussion

Table 5 lists the results for the test problems using the AGA and the results generated by using GAMS with the Xpress solver. In the second column of the table, we record the objective solution obtained by GAMS with the Xpress solver. In the third column of the table, we present the average computational time of 5 runs by GAMS with Xpress. In the fourth, fifth and sixth column of the table, we present the mean, best and standard deviation of 5 AGA runs respectively. In the seventh column of the table, we report the average computational time of 5 AGA runs. In the final column of the table, we present the percentage difference between the total cost obtained by AGA and the optimal solution found by GAMS with Xpress solver. These percentage differences are calculated by taking the ratio of the difference between the mean of the AGA's total cost and the optimal solution generated by the GAMS with Xpress solver to the optimal solution. A comparison against the optimal solution obtained by the GAMS with Xpress solver provides some measure of deviation from optimality. We note that GAMS with Xpress solver does not provide any solution for the 6 large problems, even after 50,000s of computational time. It is observed that the computational time for GAMS with Xpress solver grows drastically as the number of periods and the number of retailers increases. Therefore, it is not able to find the optimum solution in any of the large problems within the reasonable computational time.

We observe in Table 5 that the AGA can yield a good solution in short computational runtime. The percentage differences between the total cost obtained by AGA and the optimal solution obtain by GAMS with Xpress solver are less than 0.5 %. Moreover, Table 5 shows that the standard deviations over the 5 runs for the AGA are comparatively small.

Through an analysis of variance (ANOVA) tests, the output from the AGA is compared based on mean measures with respect to different levels of the number of retailers, the number of planning periods, and degree of time-dependency for the travel speed on total cost. The ANOVA test results for the main and interaction effects are shown in Table 6. ANOVA test results in Table 6 indicate that the total cost is significantly influenced by the number of retailers, the number of planning periods, and degree of time-dependency for the travel speed, suggesting that all three factors play a critical role in the inventory and routing cost.

Figure 6 indicates that the number of retailers and the number of planning periods act very much similar on the total cost.

In Fig. 6, any increase in the number of retailers or the number of planning periods lead to an increase in the total cost. The possible reason for this result is that either of them would raise the total cost. This is because the higher number of retailers and the higher number of planning periods give a higher demand. Therefore, the number of retailers and the number of planning periods significantly influence on total cost. In contrast, the impact of the degree of time-dependency for the travel speed on total cost displays a different pattern. As the degree of time-dependency for the travel speed increases, the value of the total cost decreases, as shown in Fig. 6. Since the TDIRP with time-dependency for the

Table 4 Independent factors of experimental design

Independent factors	Levels			
	1	2	3	4
The number of retailers	5	15	30	45
The number of planning periods	5	7		
Degree of time-dependency for travel speed	Low (scenarios 1)	Medium (scenarios 2)	High (scenarios 3)	

Table 5 Experimental results

Problem	GAMS with Xpress		GA				Percentage diff. (%)
	Opt. sol.	Comp. time (s)	Mean	Best	SD	Comp. time (s)	
TDIRP055-1	472.91	0.10	473.35	472.91	1.10	15.78	0.09
TDIRP057-1	589.16	0.12	590.77	589.16	0.09	15.75	0.27
TDIRP155-1	780.42	0.65	781.41	781.41	0.00	17.67	0.13
TDIRP157-1	912.56	10.18	915.41	913.67	1.24	18.97	0.31
TDIRP305-1	16,57.78	13,356.67	1,663.89	1,661.04	2.01	15.96	0.20
TDIRP307-1	1,973.35	45,128.00	1,979.37	1,975.35	3.21	15.89	0.30
TDIRP455-1	NA[a]	NA	2,183.45	2,181.23	2.10	15.67	NA
TDIRP457-1	NA	NA	3,653.78	3,651.47	2.05	17.98	NA
TDIRP055-2	450.81	0.10	451.00	450.81	0.05	16.31	0.04
TDIRP057-2	573.77	0.12	575.37	573.77	1.35	16.30	0.28
TDIRP155-2	769.67	0.67	770.05	770.05	0.00	15.02	0.05
TDIRP157-2	858.03	11.18	860.05	858.98	3.65	15.12	0.23
TDIRP305-2	1,589.17	13,608.90	1,590.13	1,589.46	0.90	15.15	0.06
TDIRP307-2	1,910.76	45,216.00	1,913.54	1,911.53	1.15	16.36	0.15
TDIRP455-2	NA	NA	1,982.71	1,981.56	0.90	16.65	NA
TDIRP457-2	NA	NA	3,458.78	3,455.67	2.50	19.77	NA
TDIRP055-3	413.53	0.10	413.55	413.53	0.00	15.35	0.00
TDIRP057-3	526.31	0.15	526.87	526.31	0.05	16.67	0.11
TDIRP155-3	721.87	0.62	721.98	721.98	0.00	15.22	0.02
TDIRP157-3	782.10	13.57	783.03	783.03	0.00	16.32	0.12
TDIRP305-3	1,322.15	14,328.90	1,322.79	1,322.99	0.08	18.75	0.05
TDIRP307-3	1,653.36	46,404.78	1,654.98	1,654.43	0.09	18.26	0.10
TDIRP455-3	NA	NA	1,895.78	1,893.67	2.01	18.87	NA
TDIRP457-3	NA	NA	3,013.78	3,012.98	1.05	18.67	NA

[a] The results are not available

travel speed has higher complexity than the IRP, we develop new chromosome structure which includes both the delivery schedule part and the vehicle routing part. In the literature, GA has been only applied to the delivery schedule part of IRPs (Abdelmaguid and Dessouky 2006; Moin et al. 2011). We also develop new crossover and new mutation operator with newly developed algorithms such as the vehicle capacity fixation, the route time fixation and partial delivery to efficiently and effectively cope with a higher degree of time dependency for the travel speed. Therefore, we can state that new data structure and new algorithm's structure mentioned in earlier sections can contribute sufficiently to obtain vehicle routing and inventory solutions in TDIRP. This can be explained by the fact that AGA performs better for higher degree of time-dependency for travel speed.

The interaction effect between the number of retailers and the number of planning periods is shown in Fig. 7. The total cost increases as the number of retailers and the number of planning periods increase. This is because the higher number of retailers and the higher number of planning periods can lead to a higher demand, under which a total cost increases.

When we consider the interaction of the degree of time-dependency for the travel speed with the number of retailers and the number of planning periods, it may be readily apparent from Fig. 7 that interactions of the number of retailers and the number of planning periods with the degree of

Table 6 ANOVA results

Source	df	SS	MS	F	*P*
The number of retailers	3	175,178,76	583,929,2	1,164.70	0.000
The number of planning periods	1	134,220,4	134,2204	267.71	0.000
Degree of time-dependency for travel speed	2	235,933	117,967	23.53	0.001
The number of retailers × the number of planning periods	3	160,274,6	534,249	106.56	0.000
The number of retailers × degree of time-dependency for travel speed	6	117,412	19,569	3.90	0.061
The number of planning periods × degree of time-dependency for travel speed	2	13,433	6,716	1.34	0.330
Error	6	30,081	5,014		
Total	23	208,596,85			

time-dependency for the travel speed have some similarities; selection of the relatively lower values for the number of retailers and the number of planning periods enables one to reduce the total cost at all degrees of time-dependency for the travel speed. The possible reason for these appearances is that either of the high number of retailers and the high number of planning periods would raise the total cost. Since the total cost is lower under high degrees of time-dependency for the travel speed as compared with the one under low degrees of time-dependency for the travel speed, the selection of relatively lower values for the number of retailers and the number of planning periods becomes crucially important to be able to reduce the total cost under low degrees of time-dependency for the travel speed. On the other hand, since a higher degree of time-dependency for the travel speed always leads to decrease in the total cost independently from the number of retailers and the number of planning periods, we can state that the developed data structure and algorithm's structure efficiently and effectively perform for solving TDIRP.

Association with manufacturing intelligence and innovation for operational excellence

The integration of the proposed AGA for solving TDIRP and the manufacturing intelligence will have a great synergy effect in supply chain. Fundamental concept of manufacturing intelligence and innovation delivers information about manufacturing processes to help businesses optimize the performance of these processes as well as manufacturing yields. However, by seamless integration of intelligent decision technologies and recent IT technology, manufacturing intelligence and innovation has been completely become the trend of logistics and supply chain. In spite of the contribution of the TDIRP to improving supply chain performance, there is a lack of research that models and solves the TDIRP. This omission is of great importance, for both theoretical and empirical reasons. The proposed AGA of solving the TDIRP needs to be embedded in various information systems for logistics and supply chain management. Therefore, this study will be a very creditable motive with which many academicians and practitioners would integrate the proposed AGA for solving TDIRP and the manufacturing intelligence.

Conclusion

Both transportation and inventory decisions ought to be made concurrently in the logistic and supply chain planning as these two areas might lead to significant benefits and more competitive distribution strategies.

In this paper, an AGA approach for solving the TDIRP is developed. A new genetic representation that focuses on the delivery schedule and vehicle routing in the form of a two-dimensional matrix is designed. A suitable crossover and mutation operator is developed. An AGA operator proposed by Yun and Gen (2003) is applied.

In the AGA construction phase, a construction algorithm with an NNS heuristic is used. In the AGA improvement phase, the crossover and mutation operators are regulated by adaptive operator.

The main concern in designing the mutation operator is to develop a suitable repair algorithm that restores feasibility.

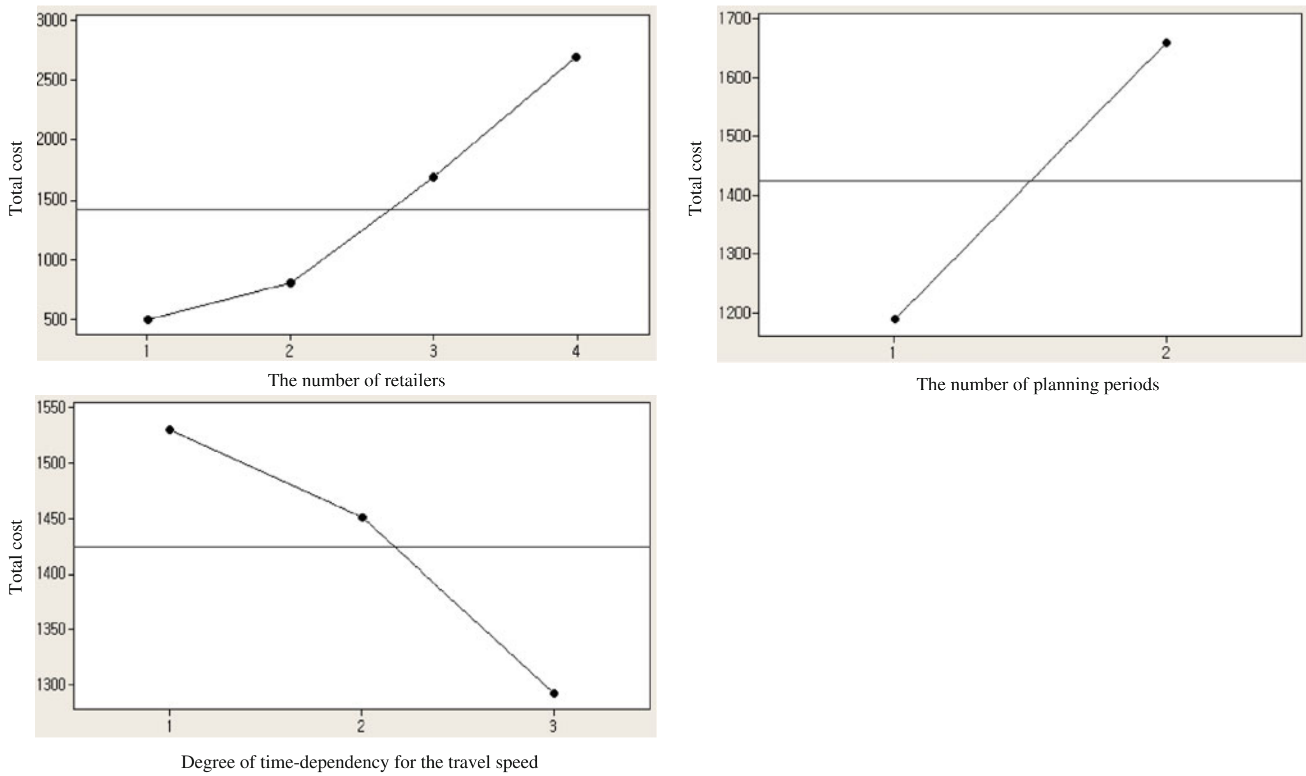

Fig. 6 Main effect of the number of retailers, the number of planning periods and degree of time-dependency for the travel speed on the total cost

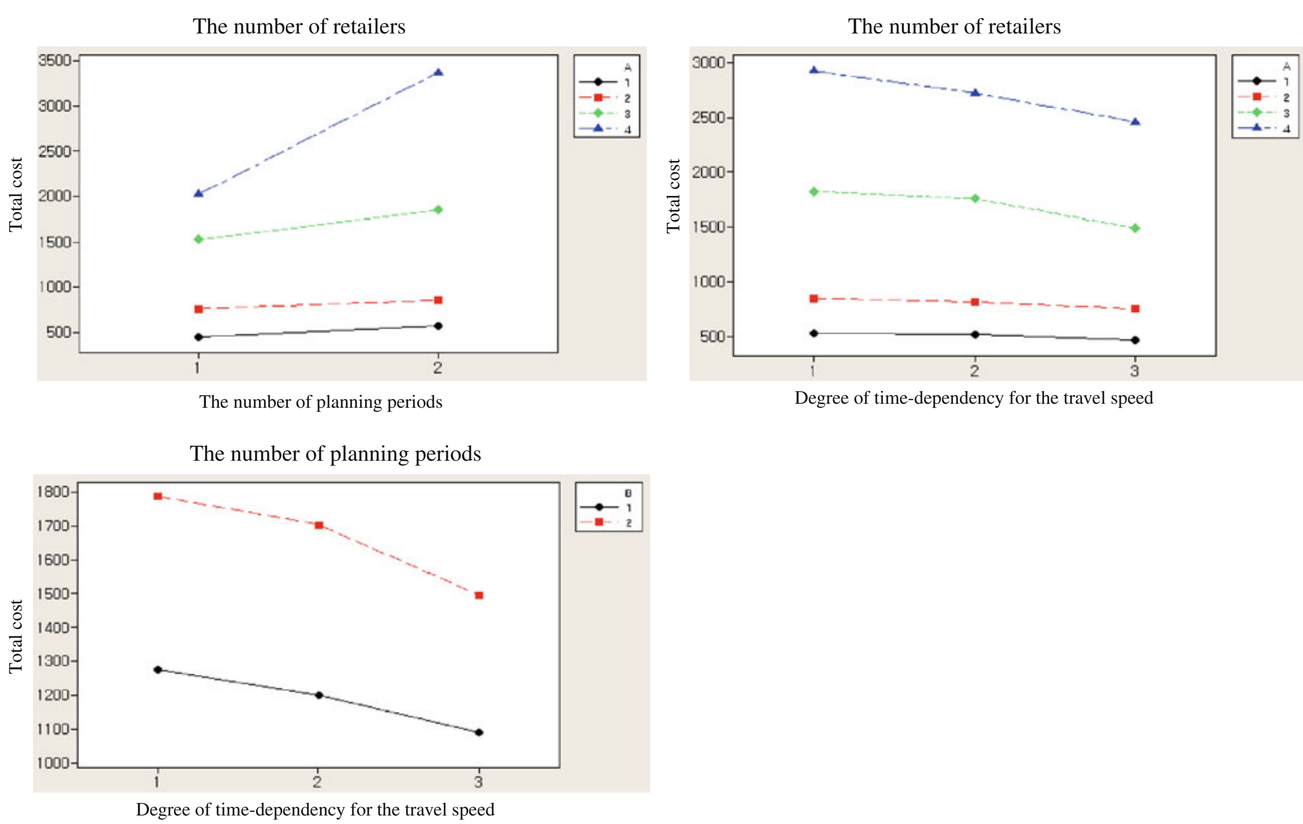

Fig. 7 Interaction effect between the number of retailers, the number of planning periods and degree of time-dependency for the travel speed on the total cost

The developed repair algorithm can provide an opportunity to save the transportation and shortage costs, and hence provide better solutions. The experimental results show the significance of the developed AGA approach.

The percentage differences between the total cost obtained by AGA and the optimal solution obtain by GAMS with Xpress solver are less than 0.5 %. Furthermore, the results were obtained with insignificantly less computational times. AGA performs relatively much better for the travel speed with the higher degree of time-dependency time.

Our research has practical implications that provide practitioners with the following managerial insights:

1. We formulated TDIRP as mixed integer programming. This study, to the best of our knowledge, is the first IRP with time dependent travel speeds. Therefore, it is expected that this research will further motivate researchers to extend our work in this area.
2. The computational results show the significance of the developed AGA. On average, AGA generates solutions that are within 0.5 % from the optimal solution. As a result, its greatest value is that it can help supply chain managers to manage inventory and distribution systems more efficiently.

For a more meaningful investigation of the time dependent inventory routing problem, one may need to extend this research to cases of more complicated time dependent IRPs. The studied problems and the developed AGA based heuristics can especially provide interesting insights for solving other problems, especially in a logistics and supply chain where the travel speed of a vehicle is not constant.

Acknowledgments This work was supported by the research fund of Hanyang University (HY-2011-P).

References

Abdelmaguid, T. F., & Dessouky, M. M. (2006). A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research*, *44*(21), 4445–4464.

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

Atamturk, A., & Savelsbergh, M. W. P. (2005). Integer-programming software systems. *Annals of Operations Research*, *140*(1), 67–124.

Baita, F., Ukovich, W., Pesenti, R., & Favaretto, D. (1998). Dynamic routing and inventory problems: A review. *Transportation Research Part A*, *32*(8), 585–598.

Balseiro, S. R., Loiseau, I., & Ramonet, J. (2011). An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers & Operations Research*, *38*(6), 954–966.

Bertazzi, L., Palettta, G., & Speranza, M. (2002). Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, *36*(1), 119–132.

Bell, W., Dalberto, M., Fisher, M., Greenfield, A., & Jaikuma, R. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interface*, *13*(6), 4–23.

Campbell, A., Clarke, L., & Savelsbergh, M. (2002). Inventory routing in practice. In D. Toth & D. Vigo (Eds.), *The Vehicle Routing Problem* (pp. 309–330). Philadelphia: SIAM.

Chan, F. T. S., Chung, S. H., & Chan, P. L. Y. (2005). An adaptive genetic algorithm with dominated genes for distributed scheduling problems. *Expert Systems with Applications*, *29*(2), 364–371.

Chan, F. T. S., Wong, T. C., & Chan, L. Y. (2009). The application of genetic algorithms to lot streaming in job-shop scheduling problem. *International Journal of Production Research*, *47*(12), 3387–3412.

Chen, H. K., Hsueh, C. F., & Chang, M. S. (2006). The real-time time dependent vehicle routing problem. *Transportation Research Part E*, *42*(5), 383–408.

Chung, S. H., Chan, F. T. S., & Chan, H. K. (2009). A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling. *Engineering Applications of Artificial Intelligence*, *22*(7), 1005–1014.

Donati, V., Montemanni, R., Casagrande, N., Rizzoli, A. E., & Gambardella, L. M. (2008). Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research*, *185*(3), 1174–1191.

Dror, M., & Ball, M. (1987). Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics*, *34*(6), 891–905.

Fogel, D. B., Fogel, G. B., & Ohkura, K. (2001). Multiple-vector self-adaptation in evolutionary algorithms. *BioSystems*, *61*, 155–162.

Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design*. New York: Wiley.

Hill, A. V., & Benton, W. C. (1992). Modeling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of the Operational Research Society*, *43*(4), 343–351.

Ichoua, A., Gendreau, M., & Potvin, J. Y. (2003). Vehicle dispatching with time dependent travel times. *European Journal of Operational Research*, *144*(2), 379–396.

Kamrani, A. K., & Gonzalez, R. (2003). A genetic algorithm-based solution methodology for modular design. *Journal of Intelligent Manufacturing*, *14*(6), 599–616.

Kuo, Y., Wang, C. C., & Chuang, P. Y. (2009). Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds. *Computers & Industrial Engineering*, *57*(4), 1385–1392.

Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering*, *59*(1), 157–165.

Liu, S. C., & Chung, C. H. (2009). A heuristic method for the vehicle routing problem with backhauls and inventory. *Journal of Intelligent Manufacturing*, *20*(1), 29–42.

Malandraki, C., & Daskin, M. S. (1992). Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science*, *26*(3), 185–200.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs* (3rd ed.). New York: Springer.

Moin, N. H., & Salhi, S. (2007). Inventory routing problems: A logistical overview. *Journal of the Operational Research Society*, *58*(9), 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. *International Journal of Production Economics*, *133*(1), 334–343.

Moon, C. U., Seo, Y. H., Yun, Y. S., & Gen, M. (2006). Adaptive genetic algorithm for advanced planning in manufacturing supply chain. *Journal of Intelligent Manufacturing*, *17*(4), 509–522.

Onwubolu, G. C., & Mutingi, M. (2003). A genetic algorithm approach for the cutting stock problem. *Journal of Intelligent Manufacturing*, *14*(2), 209–218.

Potvin, J. Y., Xu, Y., & Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, *33*(4), 1129–1137.

Richard, L., Perregaard, M., Tavares, G., Tipi, H., & Vazacopoulos, A. (2009). Solving hard-integer programming problems with Xpress-MP: A MIPLIB 2003 Case Study. *INFORMS Journal on Computing*, *21*(2), 304–313.

Tang, L., & Liu, J. (2002). A modified genetic algorithm for the flow shop sequencing problem to minimize mean flow time. *Journal of Intelligent Manufacturing*, *13*(1), 61–67.

Yang, W., Chan, F. T. S., & Kumar, V. (2012). Optimizing replenishment polices using genetic algorithm for single-warehouse multi-retailer system. *Expert Systems with Applications*, *39*(3), 3081–3086.

Yun, Y. S. (2002). Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job shop scheduling problems. *Computers and Industrial Engineering*, *43*(3), 623–644.

Yun, Y. S., & Gen, M. (2003). Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics. *Fuzzy Optimization and Decision Making*, *2*(2), 161–175.

Zandieh, M., Mozaffari, E., & Gholami, M. (2010). A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. *Journal of Intelligent Manufacturing*, *21*(6), 731–743.