Journal of Manufacturing Systems 42 (2017) 93–103

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

A genetic algorithm approach for location-inventory-routing problem with perishable products

Abdelhalim Hiassat [a], Ali Diabat [b,*], Iyad Rahwan [c]

[a] University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
[b] Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates
[c] Massachusetts Institute of Technology, Cambridge MA 02139, USA

ARTICLE INFO

Article history:
Received 20 January 2014
Received in revised form 5 October 2016
Accepted 26 October 2016
Available online 9 December 2016

Keywords:
Supply chain
Facility location
Vehicle routing
Inventory management
Perishable products
Integer programming
Genetic algorithms

ABSTRACT

In this paper, we address a location-inventory-routing model for perishable products. The model determines the number and location of required warehouses, the inventory level at each retailer, and the routes traveled by each vehicle. The proposed model adds location decisions to a recently published inventory routing problem in order to make it more practical, thus supporting the prevalent claim that integration of strategic, tactical and operational level decisions produces better results for supply chains. Given that the model developed here is NP-hard, with no algorithm capable of finding its solution in polynomial time, we develop a Genetic Algorithm approach to solve the problem efficiently. This approach achieves high quality near-optimal solutions in reasonable time. Furthermore, the unique structure of the problem requires developing a new chromosome representation, as well as local search heuristics. Finally, an analysis is carried out to verify the effectiveness of the algorithm.

1. Introduction

Researchers and practitioners often classify supply chain decisions into strategic, tactical, and operational, based on the time horizon of impact [1]. Strategic decisions have a longer time horizon of impact, which could even be years, as they deal with decisions that cannot change easily, such as the location of facilities. Tactical decisions have a time horizon of months and they include planning aspects pertaining to inventory management. Finally, operations decisions are made on a daily basis, with almost immediate impact and they include distribution decisions.

Historically, these decisions are treated separately. Each entity in the supply chain tries to minimize the cost incurred within the same facility without considering the implications of these measures on upstream or downstream entities of the supply chain. While this approach ensures that the cost for each level is minimized, the summation of all costs across the supply chain might not be at minimum. The situation where each level tries to maximize their benefit regardless of the others leads to a local optimum and sometimes sub-optimality of the systems and excessive costs [2–6], all cited in [7].

Recently, supply chain managers and researchers have realized the importance of the integration of supply chain decisions. Many researchers showed significant savings when considering a combination of the aforementioned decisions into a single model. See, for example, Nagy and Salhi [8], Wu et al. [9], Baita et al. [10], Moin and Salhi [11], Daskin et al. [12], Shen et al. [13], and Diabat et al. [7]. Many models presented in the literature combine two of the supply chain decisions into one single model. These models are location-inventory, location-routing, and inventory-routing models. However, few models integrate all three decisions and solve them simultaneously. In other words, location-inventory-routing models have not been studied extensively.

In general, one might argue about the applicability of integrating a strategic-level decision, such as location, with tactical- and operational-level decisions, such as inventory and routing. This argument is valid since these decisions belong to different levels of supply chain management and integrating them can increase the complexity of the resulting model. However, in the current work, we incorporate location decisions to the model developed by Le et al. [14], in the sense that we select which retailers will serve as distribution centers. More specifically, the model developed by Le et al. [14] deals with the storage and shipping of blood units between hospitals in a certain area. In this context, we aim to categorize these hospitals into two groups. The first group represents those medical centers which only receive blood units from other

* Corresponding author. Tel.: +971 2 810-9101; fax: +971 2 810-9901.
E-mail address: adiabat@masdar.ac.ae (A. Diabat).

http://dx.doi.org/10.1016/j.jmsy.2016.10.004
0278-6125/

ones (according to their respective demand). The other group represents the medical centers which receive more blood cells than their demand, store them, and ship them to other hospitals if needed. The problem studied by Le et al. [14] is an NP-hard problem and therefore the problem developed in this paper is also NP-hard.

Members of this latter group are named Distribution Centers (DCs) in the supply chain management literature. The former is the group of retailers. Hence, our model deals with already existing facilities. Location decisions in this sense represent the decision of which hospital will hold inventory of blood units to be shipped to other hospitals. Moreover, the fixed location cost in this regard is the overhead and extra managerial costs associated with the extra space and handling, rather than the cost of building a new facility from the ground up. This argument would justify integrating location decisions with inventory and routing decisions into a single model. In addition to this, the importance of integrating inventory with routing, as underlined by Bertsimas and Simchi-Levi [15] among others, and location with inventory, as underlined by Shen et al. [13], has been readily highlighted in the literature, further validating the integration not only of two aspects at a time, but of all three aspects simultaneously.

Knowing the fact that a moderate size mixed-integer program (MIP) can have tens of thousands of variables, it is challenging and perhaps impossible to find a global optimal solution in reasonable time with readily available computing resources. Thus, instead of searching for a global optimum, an approximate optimal, or near optimal, solution is sufficient for the problem at hand. The Genetic Algorithm (GA) is a stochastic optimization technique that depends on a random-based searching mechanism. Genetic algorithms have been successfully adapted in many areas to solve a large number of optimization problems, including scheduling and transportation problems.

In what follows, the integration of the location decisions into a recently published inventory-routing model for perishable products is introduced. Section 2 shows a brief literature of the research most related to our work. An explanation of how location decisions make previous models more practical and realistic is provided. Section 3 provides an overview of the model, and discusses its motivation and interpretation. The genetic algorithm used to solve this model is thoroughly explained in Section 4. The new chromosome representation and the new local search heuristic are also detailed. Numerical analysis to show the effectiveness of the model and the solution method is shown in Section 5, while the conclusions and the potential future work is discussed in Section 6.

2. Literature review

Researchers have recently realized the importance of integration of all the three components of the supply chain management into one model. Max Shen and Qi [16] modified the model by Daskin et al. [12] and proposed a stochastic model that considers the location, inventory and routing costs. They approximate the shipment from a warehouse to its customers using a vehicle routing model. They used Lagrangian relaxation to solve the subproblems. Lagrangian relaxation has also been implemented by Diabat et al. [17], who focus on developing an improved approach that is able to solve problems more efficiently, while Diabat and Richard [18] develop a nested Lagrangian relaxation approach for the integrated problem. Javid and Azad [19] present a model which simultaneously optimizes location, inventory and routing decisions without approximation. They show that the approximation by Max Shen and Qi [16] is only applicable under some restrictive assumptions. The model is formulated as a mixed integer convex program. They proposed a hybrid algorithm of Tabu Search and Simulated Annealing. Piece-wise linearization is applied to the joint location-inventory problem without capacity constraints [20] as well as with the consideration of capacity constraints [21].

Liu and Lee [22] consider a stochastic customer demand and include inventory costs in the location-routing problem. An initial solution is found by clustering the customers, based on an increasing order of their marginal inventory costs. Ambrosino and Grazia Scutellà [23] consider a four level location-routing problem. The authors also introduce inventory considerations. Both static and dynamic problem cases are treated. Furthermore, a heterogeneous fleet is allowed. Recently, Le et al. [14] combined inventory and routing components into one model. They proposed a column generation based heuristic to solve the model. They showed significant savings when using their model.

Supply chains with perishable products have been studied in various lines of research. Some researchers extended the economic order quantity (EOQ) policy for inventory models which include perishable products. For example, Giri and Chaudhuri [24] proposed an inventory model for a perishable product where the demand rate is a function of the on-hand inventory, and the holding cost is nonlinear. Moreover, Padmanabhan and Vrat [25] proposed a stock-dependent selling rate model where the backlogging function was assumed to be dependent on the amount of demand backlogged. Dye and Ouyang [26] extended their model by introducing a time proportional backlogging rate. In a different line of research, Hsu et al. [27] extended the vehicle routing problem with time-windows discussed in a number of papers (e.g. Koskosidis et al. [28] and Sexton and Choi [29]), by considering the randomness of the perishable food delivery process. However, both EOQ and VRP extensions lack the integration of inventory and transportation decisions. Hence, the problem of sub optimality is likely to arise.

The model described here is an extension of the model proposed by Le et al. [14]. While the authors account for transportation and inventory costs only, our model accounts for all three levels of integration. Adding the fixed location cost component makes the model more realistic and closer to real world application, without proportionally increasing its complexity. Moreover, the model is carefully formulated to maintain its linearity. A naive formulation would make the model non-linear and, consequently, much more complicated and harder to solve. In particular, Le et al. [14] use column generation to solve their model. This works well with small and medium sized instances. However, the addition of the location decisions, as well as the introduction of very large real life instances represent a challenge to existing exact solution methods. In fact, later work of Diabat and Al-Salem [30] addresses the joint inventory and routing problem with the help of a tabu search heuristic, while authors report that their heuristic outperforms the column generation approach for many problem instances. In light of this, we also develop a customized heuristic approach for the same model, which we expect will produce good quality results within reasonable time, thus making an important contribution.

Genetic algorithms are effective in achieving optimal or near-optimal solutions by solving optimization problems. As such, GAs have been widely implemented to solve a variety of single- and multi-objective problems in production and operations management that are combinatorial and NP hard [31]. However, few researchers have used GAs to tackle integrated models. Diabat et al. [32] study the capacitated facility location problem with risk pooling, while Ali Diabat and Deskoores [33] develop a hybrid GA to solve the integrated problem. Ali Diabat and Al-Salem [34] additionally consider environmental factors to their formulation. Hybrid heuristics have also been used, as in the case of Diabat [35] who study a vendor managed inventory system in a two-echelon supply chain, with a single vendor and multiple buyers.

Alp et al. [36] propose a genetic algorithm for a well-known facility location problem, the P-Median model. The P-Median model is a location/allocation model, which locates P facilities among n demand points and allocates the demand points to the facilities. Their algorithm generates good solutions quickly. Evolution is facilitated by a greedy heuristic. Baker and Ayechew [37] developed a genetic algorithm to solve the basic vehicle routing problem in which customers of known demand are supplied by a single warehouse. Convergence of the genetic algorithm described there has been accelerated by incorporating strategies for moving to neighboring solutions. Genetic algorithms were successfully applied to the multi-depot vehicle routing problem (MDVRP), [38] and location-allocation problem [39].

Moreover, Min et al. [40] propose a nonlinear mixed integer programming model and a genetic algorithm that can solve the reverse logistics problem involving product returns. The usefulness of the proposed model and algorithm was validated by its application to an illustrative example dealing with products returned from online sales. Lau et al. [41] use a search technique which depends on fuzzy logic to solve this problem. The role of fuzzy logic is to dynamically adjust the crossover rate and mutation rate after a number of consecutive generations. Another notable work in reverse logistics was presented by Alshamsi and Diabat [42], while Santibanez-Gonzalez and Diabat [43] employ Benders decomposition to solve a reverse supply chain problem. On the other hand, Al-Salem et al. [44] use piece-wise linearization for a closed-loop supply chain problem.

In this paper, we propose a genetic algorithm approach to solve the model efficiently and effectively. The complicated combinatorial model described below is extremely hard to solve using exact techniques. As such, a powerful heuristic like the one developed here is needed to find an optimal or near optimal solution in reasonable time. The contribution of our work includes both the extension of an existing formulation to account for location decisions, by integrating them with inventory and routing, as well as the development of a customized heuristic technique to solve the proposed model.

3. Model formulation

3.1. Model description

The model described in Hiassat and Diabat [45] considers the distribution of a single product from a single manufacturer to a set of retailers, I, through a set of warehouses that can be located at various predetermined sites, W. The retailers have deterministic demand but these demands may vary from one time period to the next. The goods are distributed by a homogeneous fleet of vehicles of identical capacity. In their model, it is assumed that products are perishable (i.e. have a specific shelf-life). The shelf-life is measured by the number of time periods. Also, it is assumed that out-of-stock situations never occur. Inventory holding costs are assumed to vary slightly across time. The inventory holding cost for warehouses is assumed to be the same for all candidate warehouses and therefore can be neglected. Inventory levels at retailers are limited by two constraints, namely: physical capacity at retailer sites and the shelf-life of products. It is assumed that perishability dominates the physical capacity of retailer sites in defining the upper bound inventory level. Consequently, the upper bound inventory level at customer sites is defined solely by the perishability constraints.

A feasible route is defined as a route which begins at a candidate warehouse, visits a number of retailers, and returns to the same warehouse. As such, a feasible route does not involve visiting more than one warehouse. Consequently, the number of possible feasible routes will be $W \cdot 2I$, where W and I are the number of warehouses and retailers, respectively. Feasible routes and the associated parameters (as shown later) are required as inputs to the model and, therefore, are generated prior to solving the model. The vehicle capacity is larger than the maximum customer demand at any time period. Moreover, in any time period, each vehicle travels on at most one route, and each customer is visited at most once.

Three major cost components are considered in the objective function of the model. They are as follows:

(i) warehouse fixed-location cost: the cost of establishing and operating a warehouse;
(ii) retailer unit-inventory holding cost: the cost of storing products at a retailer; and
(iii) routing cost: the cost associated with delivering the goods from a warehouse to retailers.

3.2. Notation

The notation adopted in the current formulation is described in the current section.

3.2.1. Sets

$W \triangleq$ Set of candidate warehouses, $W = 0, \ldots, |W|$
$I \triangleq$ Set of retailers, $I = 0, \ldots, |I|$
$V \triangleq$ Set of nodes, $V = W \cup I$
$T \triangleq$ Set of time periods, $T = 0, \ldots, |T|$
$R \triangleq$ Set of all feasible routes
$K \triangleq$ Set of homogeneous vehicles, $K = 0, \ldots, |K|$

3.2.2. Parameters

$f_w \triangleq$ Fixed cost of opening and operating warehouse a candidate location $\omega \in W$
$C \triangleq$ Vehicle capacity
$\tau_{max} \triangleq$ Maximum shelf-life
$d_{it} \triangleq$ Demand of customer $i \in I$ in time period $t = 1, \ldots, T, \ldots, T + \tau_{max} - 1$
$u_{it} \triangleq$ Upper bound inventory level at customer $i \in I$ at time $t \in T$, $u_{it} = \left(\sum_{\tau > t}^{\tau < t + \tau_{max}} d_{it} \right)$
$h_{it} \triangleq$ Inventory holding cost of customer $i \in I$ at time $t \in T$
$I_{i0} \triangleq$ Inventory level at customer $i \in I$ at the beginning of time period $t = 1$
$y_{it} \triangleq$ Quantity to deliver to customer $i \in I$ at time $t \in T$, $y_{it} = \sum_{r \in R} a_{ir} a_{irt}$

$$a_{ir} = \begin{cases} 1, \textit{ if } r \in R \textit{ visits customer } i \in I \\ 0, \textit{ otherwise} \end{cases}$$

$$\beta_{rw} = \begin{cases} 1, \textit{ if route } r \in R \textit{ visits customer } \omega \in W \\ 0, \textit{ otherwise} \end{cases}$$

$c_r \triangleq$ Transportation cost of route $r \in R$

3.2.3. Decision variables

$I_{it} \triangleq$ Inventory level at customer $i \in I$ at the end of time period $t \in T$
$a_{irt} \triangleq$ Quantity to deliver to customer $i \in I$ using route $r \in R$ at time $t \in T$

$$\theta_{rt} = \begin{cases} 1, \textit{ if route } r \in R \textit{ is selected at time } t \\ 0, \textit{ otherwise} \end{cases}$$

$$m_w = \begin{cases} 1, \textit{ if warehouse is opened at location } w \in W \\ 0, \textit{ otherwise} \end{cases}$$

3.3. The model

The MIP formulation of the problem can be stated as follows:
Minimize

$$Total\ Cost = \sum_{w \in W} f_w m_w + \sum_{t \in T} \left(\sum_{r \in R} c_r \theta_{rt} + \sum_{i \in I} h_{it} I_{it} \right) \quad (1)$$

Subject to

$$\sum_{r \in R} a_{ir} \theta_{rt} \le 1 \quad \forall i \in I, t \in T \quad (2)$$

$$\sum_{i \in I} a_{irt} \le C\theta_{rt} \quad \forall r \in R, t \in T \quad (3)$$

$$I_{it-1} - \sum_{r \in R} a_{ir} a_{irt} = d_{it} + I_{it} \quad \forall i \in I, t \in T \quad (4)$$

$$I_{it} \le u_{it} \quad \forall i \in I, t \in T \quad (5)$$

$$\theta_{rt} \le \sum_{w \in W} \beta_{rw} m_w \quad \forall r \in R, t \in T \quad (6)$$

$$\sum_{r \in R} \theta_{rt} \le |K| \quad \forall t \in T \quad (7)$$

$$\theta_{rt} \in \{0, 1\} \quad \forall r \in R, t \in T \quad (8)$$

$$m_w \in \{0, 1\} \quad \forall w \in W \quad (9)$$

$$a_{irt}, I_{it} \ge 0 \quad \forall i \in I, r \in R, t \in T \quad (10)$$

The objective function (1) shows the sum of costs included in this model. The first term represents the cost of opening and operating the selected warehouses, whereas the second term represents the transportation cost and the inventory holding cost. Constraints (2) guarantee that a customer is visited once at most in any time period. Constraints (3) ensure that vehicle capacities are not exceeded. Inventory balance equations are represented in constraints (4). Constraints (5) ensure that the inventory level at a customer never exceeds the total demand in the next ($\tau_{\max}$) consecutive time periods. Constraints (6) guarantee that routes start and end with open warehouses only. Constraints (7) limit the maximum number of routes at any time period to be no greater than the number of vehicles. Finally, constraints (8) and (9) restrict θ and m to be binary, and constraints (10) ensure that quantities to be shipped to customers and inventory levels are non-negative.

4. Genetic algorithm

The Genetic Algorithm (GA) is a powerful search technique. It is implemented here to solve the model described above. Fig. 1 below shows the flowchart of the algorithm used to find an optimal or near optimal solution. The algorithm starts by creating an initial population by randomly generating feasible solutions. Then, the sets of chromosomes are passed through a self-development heuristic. The fitter individuals are then chosen to evolve through crossover and mutation. This process is repeated until an exit criterion is met.

4.1. Chromosome representation

Representing the chromosome is not a trivial task in such problems. Each chromosome must carry information about warehouse location, retailers' allocation to warehouses, and routing on each time period.

Since most crossover and mutation operators in the literature assume a fixed length of each chromosome, variable length chromosomes were not used. Fig. 2 shows a sample of the encoding procedure used in this paper. The example refers to a chromosome of 2 warehouses (A and B) and 6 retailers, operating in a time span of 2 time periods. Each chromosome is divided into two parts.

The first part represents the warehouse location and allocation decisions. This part consists of $W \cdot T$ genes. Each W genes correspond to a respective time period. The location of the gene is an index corresponding to a candidate warehouse. The value of that gene is the number of retailers assigned to the warehouse in that particular time period. Warehouse A is open and is serving 4 retailers in time period 1. Also, two retailers are assigned to warehouse B in the same time period. A value of zero means that no retailer is assigned to this warehouse at this time period. In the second time period, all 6 retailers are assigned to warehouse A and none to warehouse B. However, this does not mean that warehouse B is closed (not used). Only if all gene values which correspond to the same warehouse across all time periods are zero, is the warehouse assumed closed. In that case, the cost of opening and operating that warehouse is not incurred.

The second part of the chromosome consists of $I \cdot T$ genes. For each time period, I retailers are located in I positions. The value of the gene represents the retailer number. The location, however, corresponds to the assignment to a warehouses as well as to the routing precedence. The precedence which vehicles will follow in the first time periods for warehouse A is A-3-2-4-6-A. However, vehicle capacity should be respected. As such, if the amounts shipped to 3, 2, and 4 are enough to fill up the vehicle, then two vehicles are needed and one would go the route: A-3-2-4-A, while the other would travel the route: A-6-A. In the second time period, all retailers are served by warehouse A. The number of vehicles is determined by the amount shipped and by the vehicle capacity.

4.2. Initialization

The initial population is generated in two stages. The first part of the chromosome is generated by assigning a random number between 0 and I to the first warehouse, and then iteratively calculating the remaining of the I customers and generating a random number between 0 and that value. This value is assigned to the next warehouse. This pattern continues for the remaining time periods.

For the second part, two methods were tested. In the first, random assignments were implemented. Random integers representing the retailers' numbers were generated and filled in each of the I positions in each time period. Already assigned numbers, however, are ignored to avoid duplicates. The second method is based on the distance between warehouses and retailers. In this procedure, each retailer is assigned to the nearest warehouse out of the available warehouse set. Whenever a retailer is assigned to a warehouse, a counter which represents the needed assignments for this warehouse is decreased by one. Once the counter reaches zero, the warehouse is removed from the available warehouse list. In generating the initial population, all individuals generated were designed to be feasible. Hence, a mechanism to repair infeasible chromosomes was not needed and, thus, was not implemented.

4.3. Optimal inventory

As explained earlier, the general procedure starts with solving an inventory model to optimality. The values of amounts shipped and total inventory cost are taken as inputs to the genetic algorithm. Specifically, the following model is applied to find the optimal values. The GA used here is hybridized with a linear program (LP) to solve the inventory problem. The solution of this LP specifies the quantities, to be shipped to retailers in each time period. Specifically, the LP is used to solve the following model.

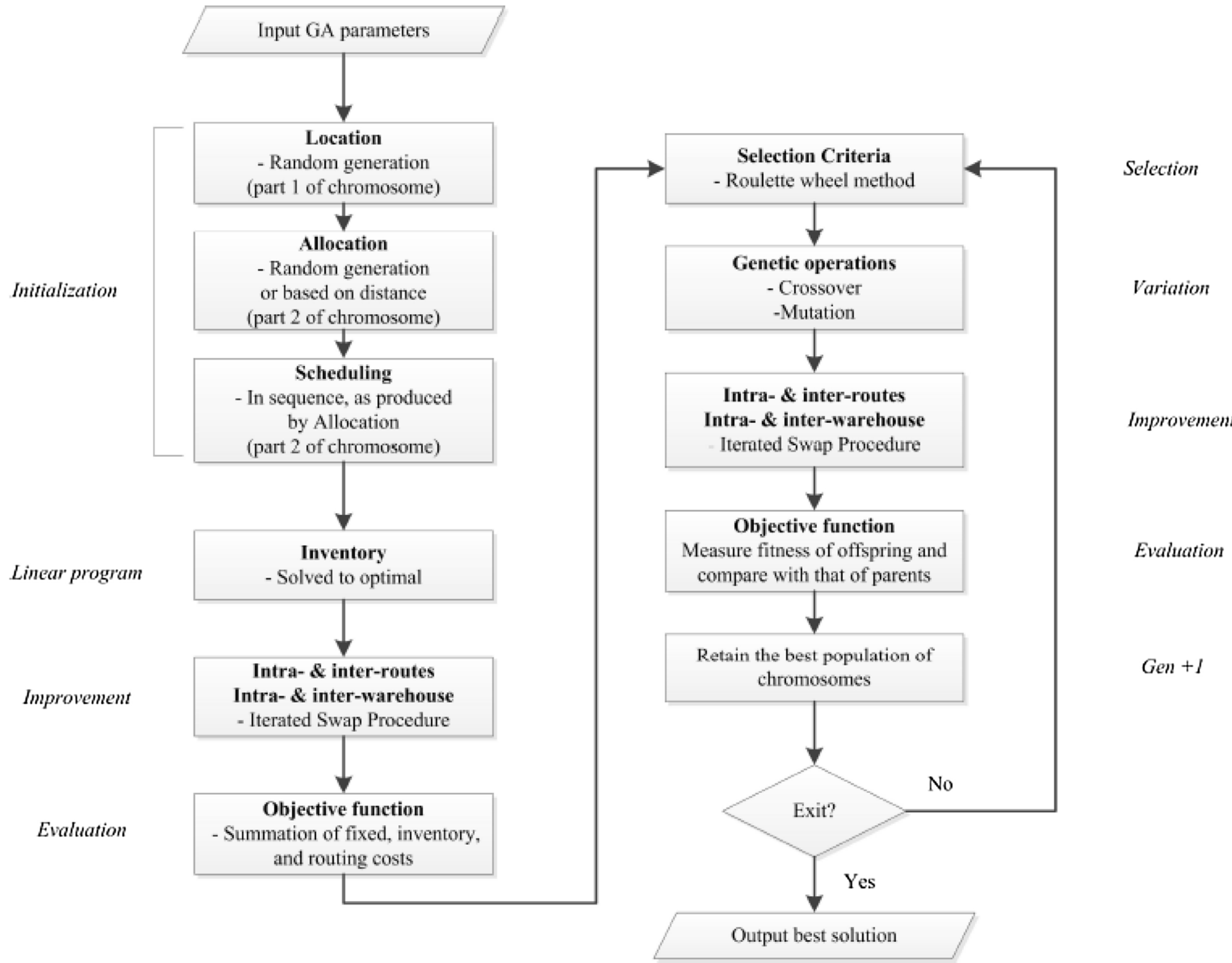

Fig. 1. Flow chart of GA.

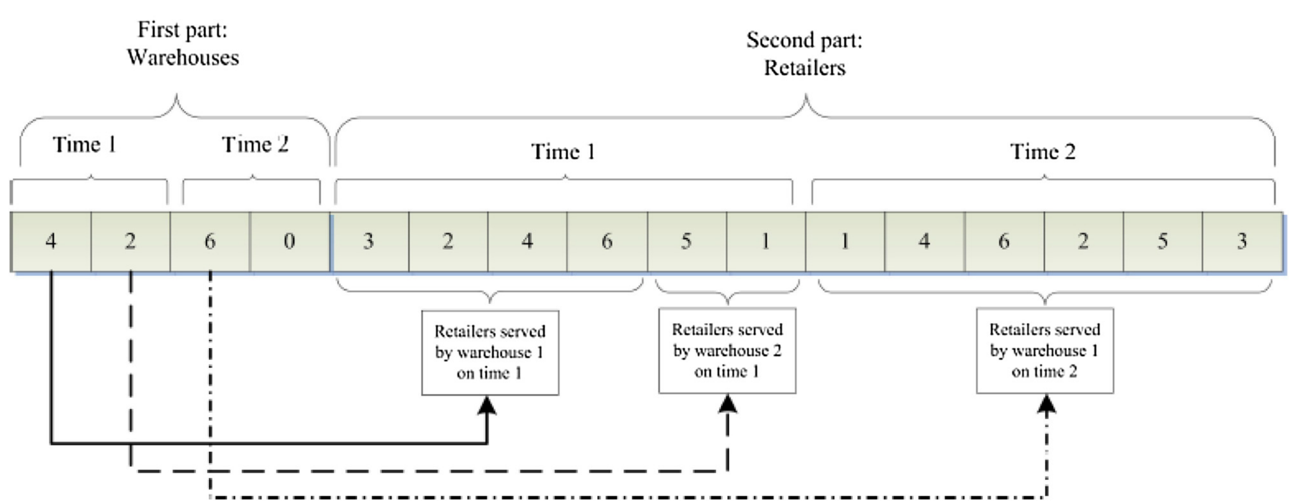

Fig. 2. Chromosome representation.

4.3.1. Decision variables

$I_{it} \triangleq$ Inventory level at customer $i \in I$ at the end of time period $t \in T$

$a_{irt} \triangleq$ Quantity to deliver to customer $i \in I$ by route $r \in R$ at time $t \in T$

4.3.2. Inventory holding sub-model

Minimize

$$Inventory\ Holding\ Cost = \sum_{t \in T} \sum_{i \in I} h_{it} I_{it} \quad (11)$$

Subject to

$$I_{it-1} - a_{it} = d_{it} + I_{it} \quad \forall i \in I, t \in T \tag{12}$$

$$I_{it} \le u_{it} \quad \forall i \in I, t \in T \tag{13}$$

$$a_{it}, I_{it} \ge 0 \quad \forall i \in I, t \in T \tag{14}$$

4.4. Improvement

The initial population and offspring generated by the genetic operations, as will be shown below, are improved using the Iterated Swap Procedure (ISP) which was originally developed by Ho. The ISP is shown in Fig. 3 below. The procedure works on the retailers' side of the chromosome as follows.

Step 1: start from the first time period, and select two genes in the retailers' side.

Step 2: exchange the positions of these two genes to form one offspring.

Step 3: swap these two genes with their neighbors to form four additional offspring.

Step 4: randomly select two genes from the next time period and go back to Step 2.

Step 5: evaluate all generated offspring and compare to parent. The best chromosome is taken, and the rest are discarded.

The ISP may exchange genes between retailers assigned to the same warehouse or between different warehouses (intra- or inter-warehouse improvement). Also, the ISP may exchange between retailers within the same route or in two different routes (intra- or inter-route improvement). This interchange, and deciding which of these changes should occur, is governed by the choice of the two genes at the start of the procedure.

4.5. Evaluation

The fitness function for our problem is the summation of all cost components involved in the model. This includes the cost of opening and operating a warehouse, known as the fixed cost (FC), the cost of shipping products in terms of cost of routes followed, or routing cost (RC), and the cost of holding products in inventory (HC), known as inventory cost.

The cost of warehouse routing is the sum of cost of all routes for a given warehouse. The sum of such costs over all warehouses constitutes the total routing cost of the chromosome. Let RCh(w) be the total delivery cost needed for warehouse w in chromosome h. If *mc* is the number of customers in route *r* and *mr* is the number of routes in warehouse *w*, then

$$RC_h(w) = \sum_{r=1}^{m_r} \left\{ c\left[v(m_c), v(0)\right] + \sum_{i=1}^{m_c} c\left[v(i-1), v(i)\right] \right\}$$

where $v(i)$ is the location of retailer i, $v(0)$ is the location of the warehouse w, $(w = 1, 2, \ldots, W)$, and

$$c(a, b) = p\left(\sqrt{(X_a - X_b)^2 + (Y_a - Y_b)^2}\right)$$

is the cost of traveling from point *a* to point *b*, where *p* is a cost factor per unit distance. Therefore, the total routing cost for chromosome *h* is

$$RC_h = \sum_{w \in W} RC_h(w)$$

And the total cost of all components is

$$X_h = FC_h + HC_h + RC_h$$

4.6. Selection

The roulette wheel selection operation is used to choose which chromosome is selected for genetic operations. The idea of this method is fairly simple. Each chromosome in the population is given a probability to be chosen. This probability is proportional to its fitness. The fitter the chromosome, the higher the probability of being selected. However, as the definition of probability suggests, there is no guarantee for any chromosome to be selected.

An important point that needs to be stressed here is that the values calculated by the fitness function described above are based on the total cost. Fitter chromosomes have lower cost values. Hence, lower costs mean higher probability to be selected. The procedure of selection commences with the calculation of the total fitness *F* of a population of size *popsize* as follows:

$$F = \sum_{h=1}^{popsize} X_h$$

The selection probability p_h for each chromosome *h* is:

$$p_h = \frac{F - X_h}{F \times (popsize - 1)}$$

Then, a random number *r* is generated in the range (0,1]. If $q_{h-1} < r \le q_h$, then chromosome *h* is selected.

4.7. Maintaining feasibility

There are plenty of algorithms proposed in the literature that allow for infeasible solutions to be part of the population and evolve from one generation to the next. This would help in finding new feasible regions and approaching a feasible optimal solution from different directions. This is useful especially when problems have disconnected feasible regions or the feasible optimal solution is at the edge of the feasible region. However, the final solution that the algorithm would converge to must be feasible. As such, techniques for controlling infeasible chromosomes must be implemented. Also, deciding how to compare infeasible solutions with feasible ones is also a major concern.

However, other algorithms maintain all-feasible populations all the time. This is accomplished by careful generation of initial population, as well as choosing the right variation operators, i.e. crossover and mutation.

In this work, only feasible chromosomes are generated and kept throughout the evolution process. Genetic operations were carefully chosen to maintain feasibility of offspring and avoid the need for a repair strategy, which can be computationally intensive and increase overall time.

4.8. Genetic operations

The search progress is achieved by two wide operations, namely crossover and mutation. Both operations constitute the exploitation and exploration part of the search. The two parts of a chromosome are different in the sense that repetition of a given time period of a value is allowed in the first part (warehouse-retailer assignment), but not in the second part. As such, genetic operations were carefully designed to maintain chromosome feasibility.

4.8.1. Crossover

Crossover is the process whereby two chromosomes (called parents) partially contribute characteristics to a new chromosome (child). The type of crossover used here is single-point crossover. The child inherits all genes from one parent up to the crossover

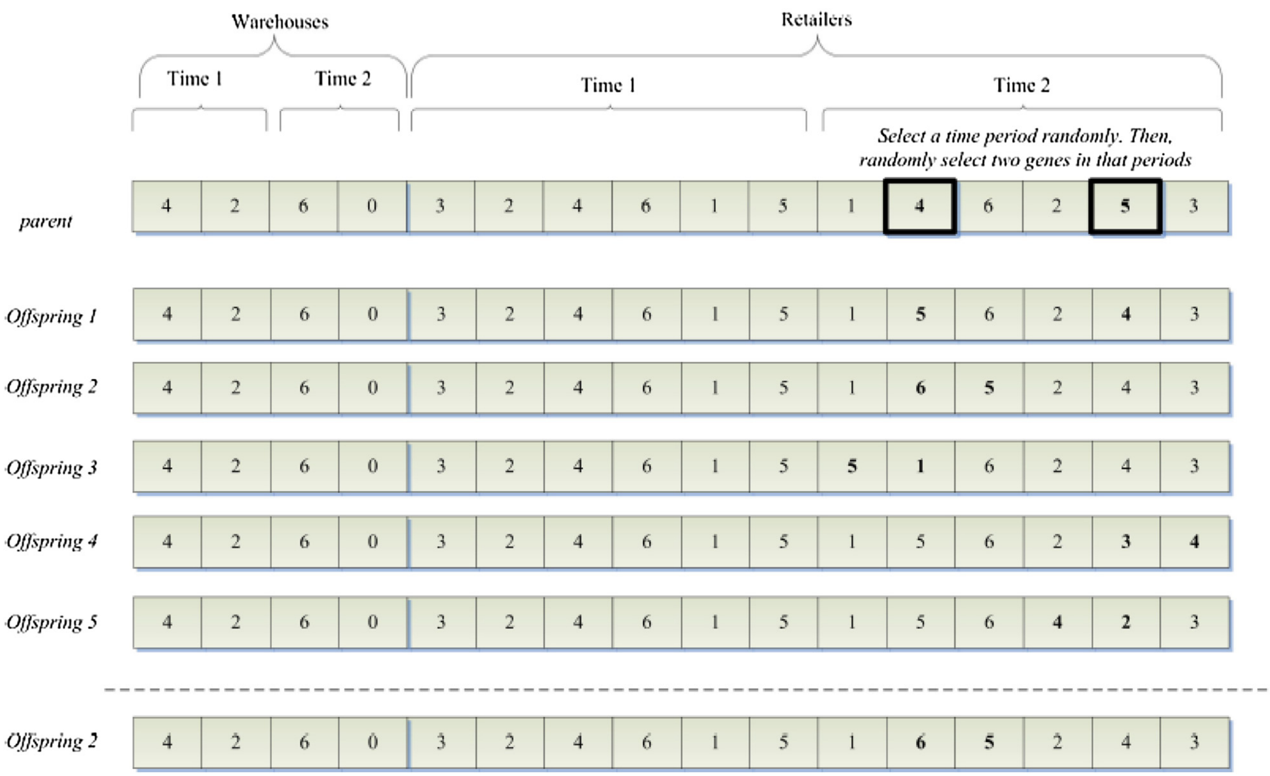

Fig. 3. The iterated swap procedure.

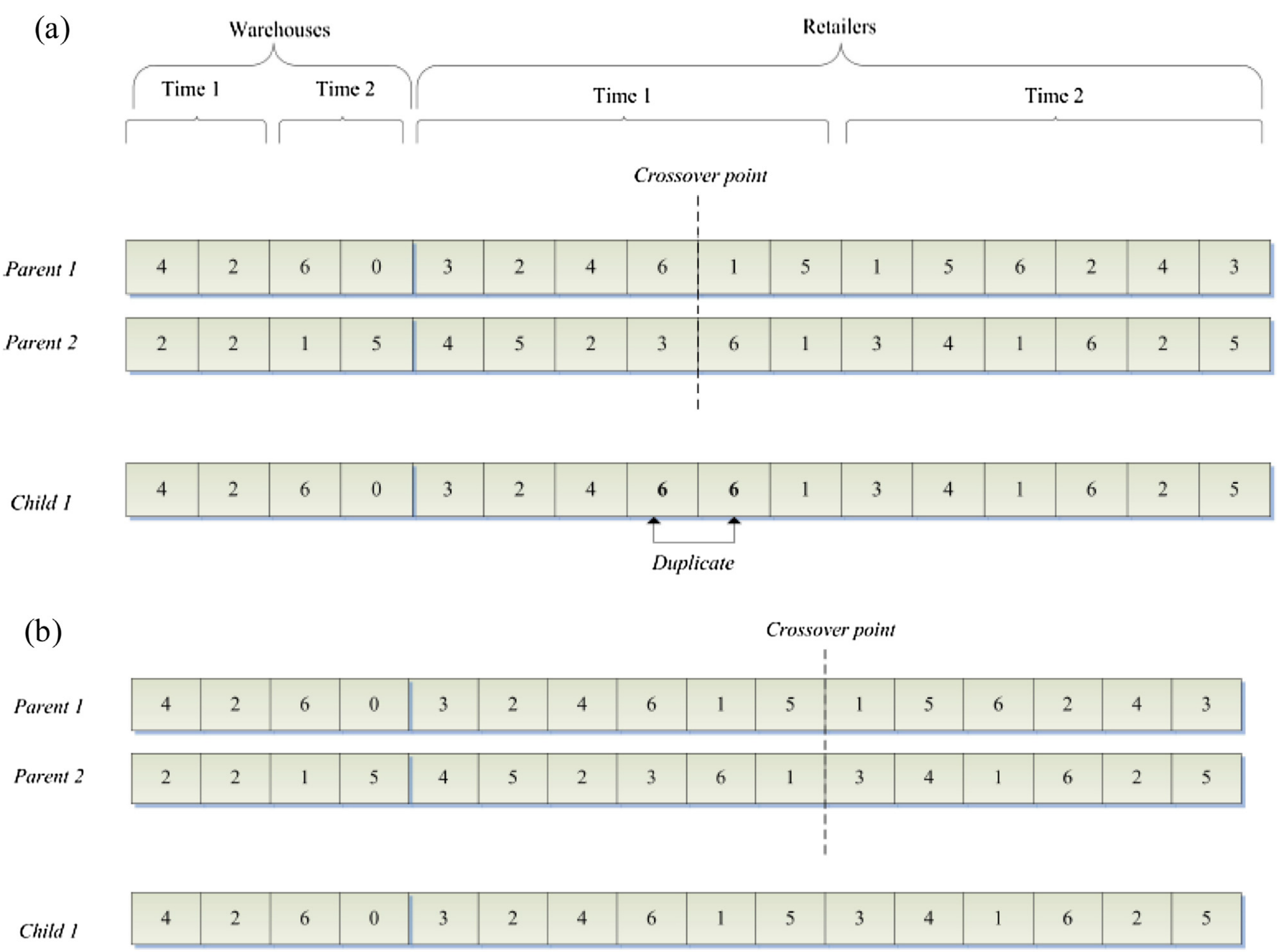

Fig. 4. Crossover operation: (a) Illegal crossover and (b) Legal crossover.

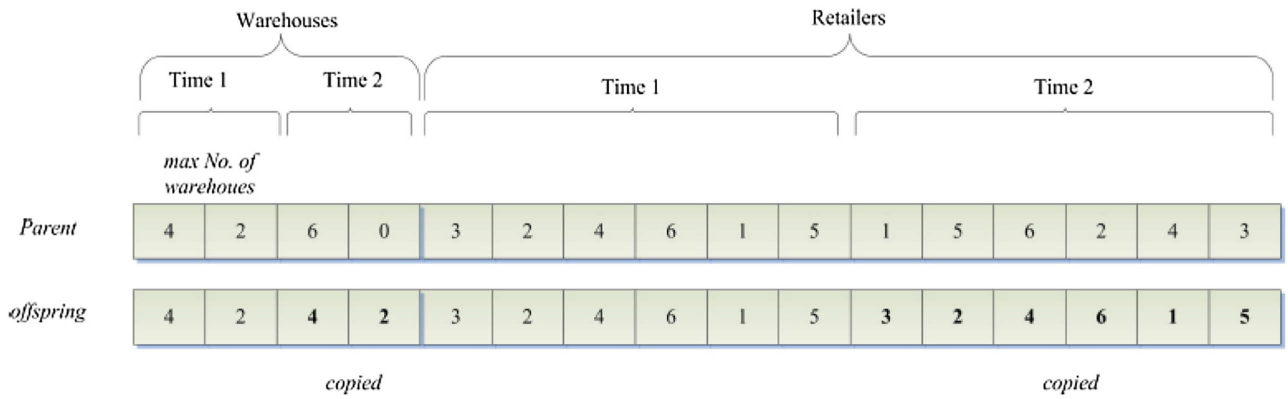

Fig. 5. Max-warehouse mutation.

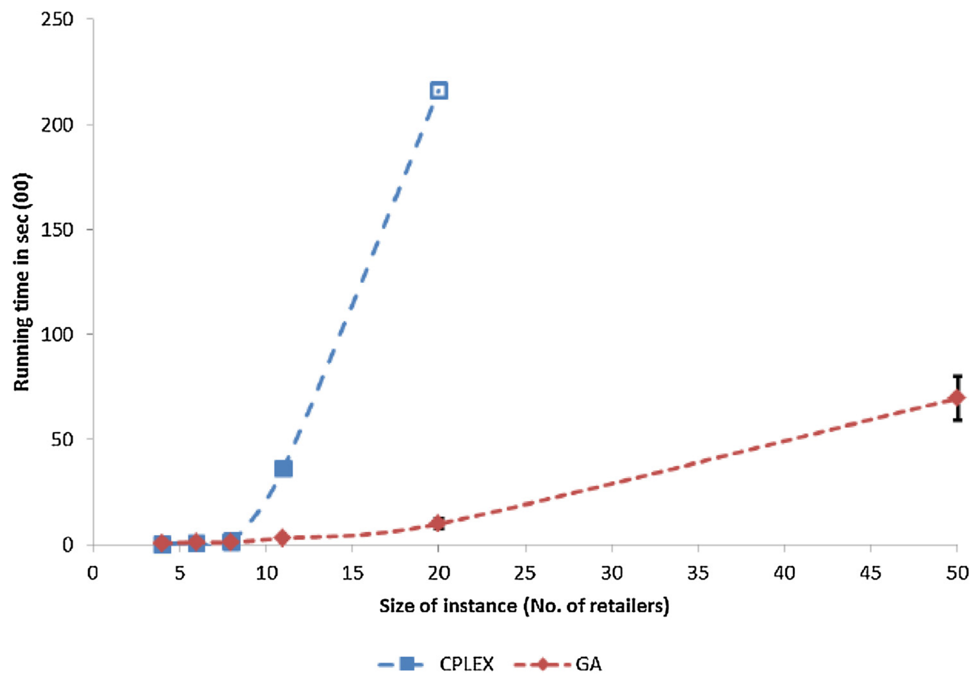

Fig. 6. Comparison between running times for CPLEX and the proposed GA, for various problem sizes.

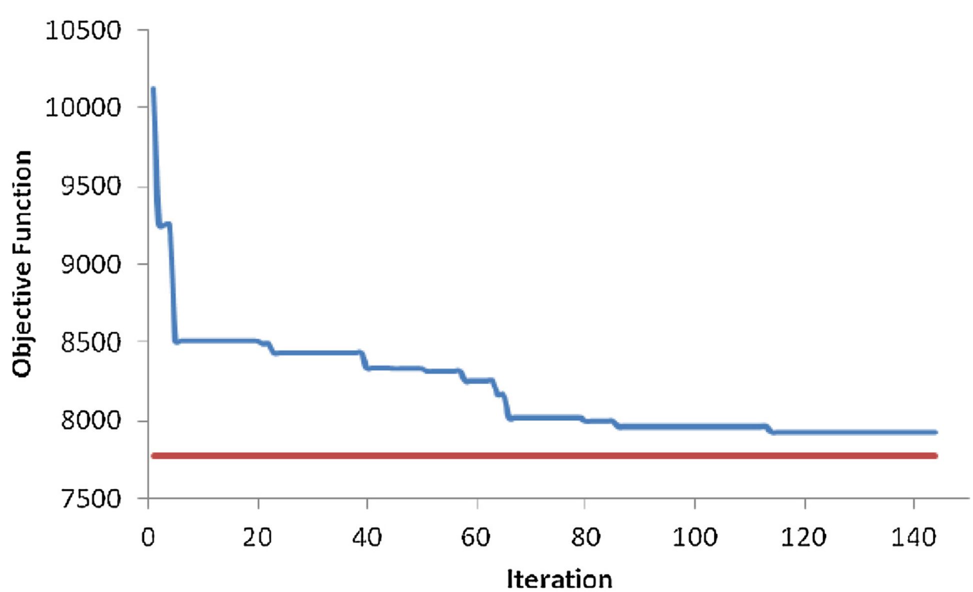

Fig. 7. Improvement profile for typical run for proposed GA.

Table 1
GA parameters for testing of different sizes.

Parameter	Value
Population Size	100
No. of Generations	1000
Idle Population	30
Crossover Rate	0.3
Mutation Rate	0.2
Crossover Type	Segment, single point
Mutation Type 1	Randomize L-A
Mutation Type 2	Max WH
Mutation Type 3	Swap

Table 2
Convergence time (in sec) for CPLEX (exact) and proposed GA (heuristic) for various instances.

No. of retailers	CPLEX	GA	Standard deviation
4	20	47	2.1
6	60	105	20
8	120	108	15
11	3600	289	86
20	21,600	983	203
50	–	6950	1021

point, and the remaining genes from the other parent. The crossover point is selected from a legal crossover set. This set consists of the boundary point between each time period both in the parts of warehouses and retailers. This is important to keep the offspring feasible. Fig. 4 below shows an example of an illegal and legal crossover.

4.8.2. Mutation

Mutation is used to explore new solutions in the solution space. It is important to avoid trapping in local optima. Many kinds of mutation operations exist and two different mutations have been tried and tested in this algorithm, as presented below:

4.8.2.1. Swap mutation. The process by which two genes are selected at random in a time period, and then they are swapped (by value) in all time periods. For example, if retailers 4 and 6 are selected, then their positions are swapped in all time periods. Using swap mutation, permutation is kept and perturbation is achieved.

4.8.2.2. Maximum warehouses open. The case where the time periods with the most warehouses open are reselected and the allocation plan is transferred to all other time periods. This mutation was developed by the authors after the observation that the GA tends to select chromosomes with fewer open warehouses. The idea is that if the chromosome is forced to have more warehouses opened, it will find its way to arrange retailers in such a way to find a better solution. This mutation has a variant where retailer sequence is also copied to other time periods (Fig. 5).

5. Results analysis

In order to show the effectiveness of the proposed algorithm, a number of randomly generated instances were tested. In particular, the data was generated as follows:

Number of retailers: 8 with 3 warehouses and 50 with 5 warehouses

Location of retailers and warehouses expressed as (X_v, Y_v) coordinates for node v: random integer in the interval [0,500]

Demand of each retailer per time period: random integer in the interval [10,100]

Inventory holding cost per time period: random integer in the interval [4.5, 5]

Capacity of fleet of vehicles: 110% of total demand of retailers

Number of vehicles: $\frac{1.1}{C}\left\{\sum_{i\in I}\sum_{t\in T} d_{it}\right\}$

Initial inventory I_{i0}: random integer in the interval $[0, d_{i1}+d_{i2}]$

Number of time periods: 5

Shelf life τ_{max}: 2 time periods

Vehicle capacity: 1.5 d_{max}

A major advantage of search heuristics such as GAs is their ability to solve medium- and large-sized instances in reasonable time. To test its effectiveness, the GA developed here is evaluated with instances of different sizes. Table 1 summarizes the GA parameter set for the genetic algorithm used for testing. A summary result is shown in Table 2.

Fig. 6 demonstrates graphically the benefit of using the GA to solve the models described in this work. The graph consists of two lines. The first line connects the running times resulting from running the genetic algorithm on different sized instances. The other line connects the running time of running the CPLEX solver in GAMS. The dashed part of this line is an estimate. The dashed line stopped at a certain point since the generation of parameters on the machine we used became impossible.

The genetic algorithm is not very efficient in finding the solution for small instances. However, as it can be seen, the genetic algorithm is superior in finding the solutions of medium and large instances.

It was noted that the exit criterion dominated with small instances sizes (<10 retailers) is the idle population cut off. As such, the maximum generation was rarely reached. Medium and large size instances, however, require more computation and, hence, the dominated exit criterion is the maximum generation. This means that with fixing the maximum generation value, the time needed for the algorithm to converge for even larger sizes is not expected to increase rapidly. It would rather level up and saturate. The slight increase, if any, in the time would be attributed to the extra time needed for processing bigger chromosomes.

Finally, after testing the parameters and choosing a suitable set of values, the quality of the solution resulted from the genetic algorithm approach can be examined. The genetic algorithm parameters were tuned to the values in Table 2, while a summary of results is provided in Table 3. The Diff. column represents the percentage difference between our solution and that resulting from applying the CPLEX solver in GAMS. The first row represents the result of applying Model 3.3 in GAMS. The second results when restricting the inventory values to be optimal, i.e. setting the variables *Iit* to certain values which correspond to the values found when solving the inventory sub-model 4.3.2. The last value is the average of 30 runs of the GA described earlier.

It is possible to observe that the gap is high for small instances. With medium-sized instances, the gap decreases. This is due to optimizing the inventory in the GA procedure. With larger instances, the effect is suppressed due to other high costs. However, the gap is expected to increase again with large-sized instances as a result of the inevitable existence of many local optima in such large solution spaces.

Furthermore, Fig. 7 shows the monotonic improvement of a typical run for the genetic algorithm. As it can be seen, the algorithm successfully improves the best value achieved, as well as the population's quality. The improvement at the beginning is significant and quick. However, as it approaches the optimal solution, it starts to slow down and saturates up to the point where no further improvement is possible. At that point, the algorithm breaks, and the procedure terminates. The lower line represents a lower bound on the optimal value of the objective function.

Table 3
Comparison of GA solution to other solvers.

	4Ret2WH		6Ret2WH		8Ret3WH	
Solver	Best Value	Diff	Best Value	Diff	Best Value	Diff
GAMS	6120		6172		7779	
GAMS-Optimal Inv	7011	14.6%	6717	8.8%	8297	6.7%
GA	7011	14.6%	6709	8.7%	8149	4.8%

6. Conclusions

The model described in this paper integrates the three levels of decisions in the supply chain, by adding a strategic level component into a recently published inventory-routing model. This integration does not pose a problem, thanks to the structure of the problem. This is due to the fact that incorporating the location decision is a mere consideration of whether a certain retailer will act as a distribution center The model is more realistic than previous models and captures real life circumstances more effectively. The numerical analysis shows that the location, allocation, inventory, and routing decisions are all affected by this integration. This model proved to show significant cost savings compared to the previous model.

Possible extensions include using multiple products, using varying shelf life time for products (depending of time of the year or temperature), and accounting for the cost of carbon emissions, which the authors are currently working on.

The genetic algorithm developed here efficiently solved medium and large instances. Solving this model using heuristics proved to be successful given the fact that exact methods do not solve very large instances in reasonable times. Different algorithm parameters have been enumerated and tested to optimize the performance of the procedure. The procedure is found to be within a small margin of errors when compared to the optimal solution for small-sized instances.

The GA is widely known and has been extensively studied. As such, many techniques and modifications can be applied to check their effectiveness on the overall performance of the algorithm. This includes trying different initialization procedures, different crossover and mutation operators and rates, and different local search heuristics. However, the GA is a search heuristic that depends on the survival of the fittest principle. The advantage of the GA is that it operates in a stochastic environment. Since models like the ones developed in this paper are highly constrained, the GA might lose this advantage. This suggests using a different search heuristic for such models. Examples of promising procedures include Ant Colony Optimization and Particle Swarm, among others.

References

[1] Miller T. Hierarchical operations and supply chain planning. New, York NY: Springer; 2002.
[2] Stevens GC. Integrating the supply chain. Int J Phys Distrib Logist Manage 1989;19(8):3–8.
[3] Gunasekaran A, Patel C, Tirtiroglu E. Performance measures and metrics in a supply chain environment. Int J Oper Prod Manage 2001;21(1/2):71–87.
[4] Cavinato JL. A total cost/value model for supply chain competitiveness. J Bus Logist 1992;13. p. 285.
[5] Chopra S, Meindl P. Supply chain management: strategy, planning, and operation. Pearson Prentice Hall; 2007.
[6] Silver E, Pyke DF, Peterson R. Inventory management and production planning and scheduling; 1998.
[7] Diabat A, Richard J-P, Codrington C. A lagrangian relaxation approach to simultaneous strategic and tactical planning in supply chain design. Ann Oper Res 2013;203(1):55–80.
[8] Nagy G, Salhi S. Location-routing: issues, models and methods. Eur J Oper Res 2007;177(2):649–72.
[9] Wu T-H, Low C, Bai J-W. Heuristic solutions to multi-depot location-routing problems. Comput Oper Res 2002;29(10):1393–415.
[10] Baita F, Ukovich W, Pesenti R, Favaretto D. Dynamic routing-and-inventory problems: a review. Transp Res Part A: Policy Pract 1998;32(8):585–98.
[11] Moin NH, Salhi S. Inventory routing problems: a logistical overview. J Oper Res Soc 2007;58(9):1185–94.
[12] Daskin M, Coullard C, Shen Z-J. An inventory-location model: formulation, solution algorithm and computational results. Ann Oper Res 2002;110(1–4):83–106.
[13] Shen Z-JM, Coullard C, Daskin MS. A joint location-Inventory model. Transp Sci 2003;37(1):40–55.
[14] Le T, Diabat A, Richard J-P, Yih Y. A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. Optim Lett 2013;7(7):1481–502.
[15] Bertsimas DJ, Simchi-Levi D. A new generation of vehicle routing research: robust algorithms, addressing uncertainty. Oper Res 1996;44(2):286–304.
[16] Max Shen ZJ, Qi L. Incorporating inventory and routing costs in strategic location models. Eur J Oper Res 2007;179(2):372–89.
[17] Diabat A, Battaïa O, Nazzal D. An improved Lagrangian relaxation-based heuristic for a joint location-inventory problem. Comput Oper Res 2015;61:170–8.
[18] Diabat A, Richard J-P. An integrated supply chain problem: a nested lagrangian relaxation approach. Ann Oper Res 2015;229(1):303–23.
[19] Javid AA, Azad N. Incorporating location, routing and inventory decisions in supply chain network design. Transp Res Part E: Logist Transp Rev 2010;46(5):582–97.
[20] Diabat A. A capacitated facility location and inventory management problem with single sourcing. Optim Lett 2015:1–16.
[21] Diabat A, Theodorou E. A location-inventory supply chain problem: reformulation and piecewise linearization. Comput Ind Eng 2015;90:381–9.
[22] Liu SC, Lee SB. A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. Int J Adv Manuf Technol 2003;22(11-12):941–50.
[23] Ambrosino D, Grazia Scutellà M. Distribution network design: new problems and related models. Eur J Oper Res 2005;165(3):610–24.
[24] Giri BC, Chaudhuri KS. Deterministic models of perishable inventory with stock-dependent demand rate and nonlinear holding cost. Eur J Oper Res 1998;105(3):467–74.
[25] Padmanabhan G, Vrat P. EOQ models for perishable items under stock dependent selling rate. Eur J Oper Res 1995;86(2):281–92.
[26] Dye C-Y, Ouyang L-Y. An EOQ model for perishable items under stock-dependent selling rate and time-dependent partial backlogging. Eur J Oper Res 2005;163(3):776–83.
[27] Hsu C-I, Hung S-F, Li H-C. Vehicle routing problem with time-windows for perishable food delivery. J Food Eng 2007;80(2):465–75.
[28] Koskosidis YA, Powell WB, Solomon MM. An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. Transp Sci 1992;26(2):69–85.
[29] Sexton TR, Choi Y-M. Pickup and delivery of partial loads with soft time windows. Am J Math Manage Sci 1986;6(3–4):369–98.
[30] Diabat A, Al-Salem M. A joint location inventory problem with environmental considerations. Int J Prod Econ 2014, in press.
[31] Dimopoulos C, Zalzala AMS. Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. Evol Comput IEEE Trans 2000;4(2):93–113.
[32] Diabat A, Aouam T, Ozsen L. An evolutionary programming approach for solving the capacitated facility location problem with risk pooling. Int J Appl Decis Sci 2009;2(4):389–405.
[33] Diabat A, Deskoores RM. A hybrid genetic algorithm based heuristic for an integrated supply chain problem. J Manuf Syst 2015;38:172–80.
[34] Diabat A, Al-Salem M. An integrated supply chain problem with environmental considerations. Int J Prod Econ 2015;164:330–8.
[35] Diabat A. Hybrid algorithm for a vendor managed inventory system in a two-echelon supply chain. Eur J Oper Res 2014;238(1):114–21.
[36] Alp O, Erkut E, Drezner Z. An efficient genetic algorithm for the p-median problem. Ann Oper Res 2003;122(1–4):21–42.
[37] Baker BM, Ayechew MA. A genetic algorithm for the vehicle routing problem. Comput Oper Res 2003;30(5):787–800.
[38] Skok M, Skrlec D, Krajcar S. The genetic algorithm method for multiple depot capacitated vehicle routing problem solving. In Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings. Fourth International Conference on 2000:520–6.
[39] Zhou G, Min H, Gen M. The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach. Comput Ind Eng 2002;43(1):251–61.
[40] Min H, Jeung Ko H, Seong Ko C. A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns. Omega 2006;34(1):56–69.

[41] Lau HCW, Chan T, Tsui WT, Pang WK. Application of genetic algorithms to solve the multidepot vehicle routing problem. Autom Sci Eng, IEEE Trans 2010;7(2):383–92.
[42] Alshamsi A, Diabat A. A reverse logistics network design. J Manuf Syst 2015;37(3):589–98.
[43] Santibanez-Gonzalez EDR, Diabat A. Solving a reverse supply chain design problem by improved Benders decomposition schemes. Comput Ind Eng 2013;66(4):889–98.
[44] Al-Salem M, Diabat A, Dalalah D, Alrefaei M. A closed-loop supply chain management problem: reformulation and piecewise linearization. J Manuf Syst 2016.
[45] Hiassat A, Diabat A. A location-inventory-routing problem with perishable products. Proceedings of the 41st International Conference on Computers and Industrial Engineering 2011.