E-ISSN: 2717-2937 | P-ISSN: 2783-1337

International Journal of Research in Industrial Engineering

www.riejournal.com

Int. J. Res. Ind. Eng. Vol. 12, No. 4 (2023) 397–413.

Paper Type: Research Paper

# A New Efficient Genetic Algorithm-Taguchi-Based Approach for Multi-Period Inventory Routing Problem

**Amin Farahbakhsh[1], Amir Saman Kheirkhah[1,*]**

[1] Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran; a.farahbakhsh@eng.basu.ac.ir; kheirkhah@basu.ac.ir.

**Citation:**

Farahbakhsh, A., & Kheirkhah, A. (2023). A new efficient genetic algorithm-Taguchi-based approach for multi-period inventory routing problem. *International journal of research in industrial engineering, 12*(4), 397-413.

Received: 11/06/2023 Reviewed: 12/07/2023 Revised: 21/08/2023 Accepted: 01/09/2023

## Abstract

The inventory routing problem arises from the combination of the vehicle routing problem and the vendor-managed inventory problem. In this paper, we present a mathematical model and a novel genetic algorithm for solving the multi-period inventory routing problem. The objective is to supply products to scattered customers within a given time horizon while managing customer inventories to avoid shortages and minimize total inventory and transportation costs. To represent solutions for this problem, we introduce a new chromosomal structure. This structure offers simplicity in encoding and decoding solutions, maintains feasibility after crossover and mutation operations, addresses both routing and inventory management in a single step, and consolidates information about each solution method comprehensively. The algorithm parameters, including crossover and mutation rates, population size, number of iterations, and selection pressure, are fine-tuned using the Taguchi method. To assess algorithm efficiency, we utilize standard instances from the literature. Our results demonstrate that the proposed algorithm performs favorably compared to previous approaches.

**Keywords:** Inventory routing problem, Genetic algorithm, Metaheuristic, Optimization.

## 1 | Introduction

Licensee International Journal of Research in Industrial Engineering. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0).

Supply chain management is a complex system of interrelated components that can significantly impact each other. Two critical components of supply chain management are transportation and inventory management [1]. In the literature, the coordination between transportation and inventory management is often referred to as the Inventory Routing Problem (IRP). The IRP aims to minimize costs by determining the optimal distribution and inventory strategy [2]. It involves three main decisions: how much and when to deliver to each customer, what routes to use for each delivery, and what inventory levels to maintain at each location [3]. The IRP is a challenging and practical problem that arises in various industries and contexts [4].

This paper focuses on the multi-period IRP, which involves determining the optimal inventory levels for each period. Additionally, this problem is classified as an NP-hard problem, meaning that exact methods cannot efficiently solve large-scale and complex instances [5]. Therefore, metaheuristic algorithms, which are approximate methods inspired by natural phenomena or human behavior, are

Corresponding Author: kheirkhah@basu.ac.ir

https://doi.org/10.22105/riej.2023.403685.1387

often employed to find good solutions for the IRP. These metaheuristic algorithms are widely used across different fields to tackle complex real-world problems [6]. One of the most popular metaheuristic algorithms is the Genetic Algorithm (GA), which mimics the process of biological evolution [7]. The GA operates on a population of solutions called chromosomes, applying operators such as selection, crossover, and mutation to improve them over generations. The key elements of metaheuristic algorithms are intensification and diversification [8]. The GA's performance depends on several parameters, including population size, crossover rate, and mutation rate [9]. In this paper, we propose a new chromosome representation for the IRP and use the Taguchi method to fine-tune the GA parameters. We tested our approach on several benchmark instances and compared it with existing methods from the literature. We demonstrate that our approach can find better or more competitive solutions for the IRP in a reasonable time.

The rest of the paper is organized as follows: Section 2 reviews the related work on the IRP and metaheuristic algorithms. Section 3 presents the mathematical model and the notation for the IRP. Section 4 describes the proposed GA and its components in detail. Section 5 reports the numerical results and the analysis of our approach. Finally, Section 6 concludes the paper and suggests some directions for future research.

## 2 | Literature Review

The first time IRP was presented by Bell et al. [10], and it has been studied by researchers from various aspects since then. The field of IRP is broad, and researchers have addressed this problem from various angles, such as time periods, transportation fleet, inventory policies, and algorithms. The multi-period inventory routing problem considering the carbon emission regulations was proposed by Cheng et al. [11]. Perishable products were studied in a multi-period inventory routing problem [12]. Xiao and Rao [13] presented the multi-product and multi-period IRP, considering the time window. Alinaghian et al. [14] presented a piecewise linearized green multi-period IRP with time windows. There are other cases that are not the subject of this research and have been discussed in detail in the review article [15].

Exact algorithms for IRP were designed by Solyalı and Süral [16], where the branch-and-cut algorithm is represented by Coelho and Laporte [17], and branch-and-price-and-cut by Andersson et al. [18]. Since exact methods are not efficient for large dimensions of the IRP, the solution to this problem was presented by Archetti et al. [19] using an efficient matheuristic algorithm. Yu et al. [20] introduced a distance-based clustering method based on an ant colony optimization approach. Cordeau et al. [21] designed a decomposition approach to tackle large-scale instances of the IRP. Su et al. [22] integrated a local search-based metaheuristic with mathematical programming. An augmented Tabu Search (TS) algorithm and a Differential Evolution (DE) algorithm for IRP were implemented by Alinaghian et al. [14]. A two-stage hybrid metaheuristic algorithm was proposed by Wu et al. [23] for a multi-period location-inventory-routing problem with time windows and fuel consumption.

John Holland [24] developed the first GA in the early 1970s. To obtain good solutions for multi-period IRP, various GA approaches have been developed [25]. A GA for the IRP with lost sales proposed in [26] uses two matrix chromosomes: the first to determine clustering and the second to determine the route. Using the GA method, Othman et al. [27] proposed simulation optimization modeling for the IRP. The stochastic periodic Can-Deliver policy, which permits early replenishment, serves as the foundation for the IRP simulation model. The chromosomal structure in this study is determined by three levels of warehouse replenishment, and the GA presents a classification of customers. Routing in each category is done by a heuristic algorithm.

Hiassat et al. [28] have developed an efficient GA approach to solve the location-inventory-routing problem with perishable products. The authors used a new chromosomal structure in their GA. Azadeh et al. [29] proposed an IRP for a single perishable product, which has been solved using a GA. The proposed algorithm parameters are tuned using the Taguchi method. In the chromosome presented in

Farahbakhsh and Kheirkhah| Int. J. Res. Ind. Eng. 12(4) (2023) 397-413

this study, the initial and final customers for the visit are determined using a matrix. However, the sequence of middle customers is not determined.

Arab et al. [30] proposed a Non-Dominated Sorting Genetic Algorithm (NSGA-II) for solving the multi-objective, multi-period, and multi-product inventory-routing problem. Yavari et al. [31] investigated solving the multi-period location-inventory-routing problem of perishable products with a GA. Amri Sakhri [32] developed a GA to solve IRP with different crossover structures. Mahjoob et al. [4] proposed a Modified Adaptive Genetic Algorithm (MAGA) to solve the multi-product and multi-period IRP. In this research, two matrix structures consisting of real and binary numbers have been used to represent chromosomes. Furthermore, the method uses a GA only for clustering, while routing is done by another heuristic algorithm. For the IRP with deterministic customer demand, Sakhri et al. [33] created a Memetic Algorithm (MA) based on GA and Variable Neighborhood Search (VNS) methods.

A review of the literature shows that the matrix structures presented in previous research are incapable of simultaneously classifying, routing, and managing inventory for the IRP. For this reason, the researchers solved the model using a two-step procedure: clustering first and then routing with VRP heuristic algorithms. Chromosome structures must contain all the necessary information about the problem. However, this issue complicates chromosome structure. The complexity of chromosome structure reduces the speed and accuracy of the GA. A simpler chromosome has a better ability to improve the performance of the GA.

In this research, a new chromosomal structure has been presented for the GA, which addresses the defects in previous algorithms for solving the IRP problem. Among the important features of this structure, the following can be mentioned:

I. Using this matrix structure, there is no need to solve the problem of routing and inventory management in two stages; both problems will be solved simultaneously.
II. All information related to each solution method, including the distribution route, order of distribution, shipping amount, stock level of warehouses, customers not covered, and unused machines, is contained in a simple way.
III. The simplicity of chromosome encoding and decoding increases the speed and accuracy of solving the problem, thereby enhancing the performance of the GA in the IRP problem.
IV. The ability to develop this structure for the Production Routing Problem (PRP) is also one of its other features.

In summary, the chromosomal structure presented for the first time has improved the performance of the GA in solving the IRP by addressing the defects of previous algorithms.

## 3 | Problem Description and Formulation

In this study, the problem is expressed as a multi-period inventory routing problem. The supplier must decide which products to transfer to customers to meet the demand specified over the finite planning horizon. The problem is periodic: in each period, the beginning inventory for customers is known, and the demand level for each customer in each period is both known and limited. These demands are specific, deterministic, but variable over time periods, and backlogging and split-delivery are not allowed. Capacitated vehicles transfer goods from the supplier to the customers and return them to the supplier. It is assumed that there is enough inventory at the supplier to satisfy all demands.

A homogeneous transportation fleet is used to respond to customer demand. The holding cost is also assumed to remain constant over the planning horizon. The transportation cost per trip consists of a fixed cost incurred on each trip and a variable cost proportional to the distance traveled. To meet customer demands, the Order-Up-to (OU) policy was adopted, which states that the quantity delivered to a customer must equal the inventory capacity policy. The objective of the problem is to minimize the total

transportation and inventory holding costs of the system while ensuring there are no unsatisfied demands from customers in each period. Decision variables in the problem include inventories for customers, material delivery levels to customers, and routes for delivering materials in each period.



The mathematical model is proposed as a single-product, multi-period, and single-objective one. The assumptions considered for modeling and evaluation are as follows:

- *Demands from customers must be met completely in each period.*
- *Distances between points are expressed as Euclidean distances.*
- *The location of suppliers and customers is fixed and determined.*
- *Customer demands are known and fixed.*
- *Each customer must be visited once, with the same vehicle, in each period.*
- *Each vehicle can be launched once in each period.*
- *Vehicle capacities and customers' inventories are limited and determined.*
- *Established routes must only begin from the supplier and end at the supplier.*
- *Customer demands remain fixed and consistent across all periods.*

## 3.1 | Notation

The notation adopted in the current formulation is described as follows:

### 3.1.1 | Sets and indices

| | |
|---|---|
| $N = S \cup C$ | Set of all nodes. |
| $S = \{0\}$ | Set of suppliers. |
| $C=\{1,2,\ldots,\lvert C\rvert\}$ | Set of customers. |
| $K=\{1,\ldots,\lvert K\rvert\}$ | Set of homogeneous vehicles. |
| $T=\{1,\ldots,\lvert T\rvert\}$ | Set of time periods. |
| $i$ & $j$ | Supplier and customers index. |
| $k$ | Vehicle index. |
| $t$ | Time period index. |

### 3.1.2 | Parameters

| | |
|---|---|
| $d_i^t$ | Demand of customer i in time period t. |
| $C_{ij}$ | Distant-dependent travel cost between customer i and j. |
| $h_i$ | Unit inventory holding cost at the place of customer i. |
| $f$ | Fixed transportation cost. |
| $Q$ | Maximum capacity of vehicles. |
| $U_i$ | Maximum inventory holding capacity of customer i. |

### 3.1.3 | Variables

| | |
|---|---|
| $y_{ij}^{kt}$ | 1, if the path from node i to node j is traversed by vehicle k in time period t, and 0 otherwise. |
| $qs_{ik}^t$ | The amount of transferred product to customer i by vehicle k in time period t. |
| $ql_{ij}^{kt}$ | The loading amount of vehicle k in route from node i to node j in time period t. |
| $I_{it}$ | Inventory level of customer i in time period t. |

## 3.2 | Model Formulation

The mathematical formulation is presented in this section. The objective function given in *Eq. (1)* minimizes the total cost. The first section of the objective function minimizes the fixed cost of the

Farahbakhsh and Kheirkhah| Int. J. Res. Ind. Eng. 12(4) (2023) 397-413

supplier's used vehicle. The second section illustrates the inventory holding cost at the customer locations. Routing and transshipment costs are represented in the third section.

$$\text{Min } z = \sum_{k\in K}\sum_{t\in T}\sum_{i\in S}\sum_{j\in C} f \times y_{ij}^{kt} + \sum_{i\in C}\sum_{t\in T} h_i \times I_{it} + \sum_{k\in K}\sum_{t\in T}\sum_{(i,j)\in N} c_{ij} \times y_{ij}^{kt}, \tag{1}$$

s.t.

$$I_i^t \le U_i \quad \text{for all } i \in C, t \in T, \tag{2}$$

$$I_i^t = I_i^{t-1} + \sum_{k\in K} qs_{ik}^t - d_i^t \quad \text{for all } t \in T, i \in C, \tag{3}$$

$$\sum_{k\in K} qs_{ik}^t \le U_i - I_i^{t-1} \quad \text{for all } t \in T, i \in C, \tag{4}$$

$$\sum_{i\in S}\sum_{j\in N} y_{ij}^{kt} \le 1 \quad \text{for all } k \in K, t \in T, \tag{5}$$

$$\sum_{i\in N}\sum_{k\in K} y_{ij}^{kt} \le 1 \quad \text{for all } t \in T, j \in C, \tag{6}$$

$$\sum_{i\in N} y_{ij}^{kt} + \sum_{i\in N} y_{ji}^{kt} = 2 \times \sum_{i\in N} y_{ij}^{kt} \quad \text{for all } t \in T, j \in N, k \in K, \tag{7}$$

$$y_{ii}^{kt} = 0 \quad \text{for all } t \in T, i \in N, k \in K, \tag{8}$$

$$\sum_{i\in C} y_{ij}^{kt} = \sum_{i\in C} y_{ji}^{kt} \quad \text{for all } t \in T, k \in K, j \in S, \tag{9}$$

$$\sum_{j\in C} ql_{ij}^{kt} = \sum_{j\in C} qs_{jk}^t \quad \text{for all } t \in T, i \in S, k \in K, \tag{10}$$

$$\sum_{i\in N} ql_{ij}^{kt} - \sum_{i\in N} ql_{ji}^{kt} = qs_{jk}^t \quad \text{for all } t \in T, j \in C, k \in K, \tag{11}$$

$$ql_{ij}^{kt} \le Q \times y_{ij}^{kt} \quad \text{for all } t \in T,\ i, j) \in N, k \in K, \tag{12}$$

$$y_{ij}^{kt} \in \{0,1\}, \tag{13}$$

$$qs_{ik}^t, ql_{ij}^{kt}, I_i^t \ge 0. \tag{14}$$

*Eq. (2)* ensures the inventory level of each customer never exceeds the maximum inventory holding capacity. The inventory balance constraint at customer i, as shown in *Eq. (3)*, is equal to the inventory level in period t-1, by adding the total quantity transshipped in period t and subtracting the demand in period t. *Eq. (4)* ensures that the quantity delivered to the customer is below the remaining customer's inventory holding capacity. *Eq. (5)* states that each vehicle is not used more than once in each period by the supplier, as *Eq. (6)* ensures that each customer is not visited more than once in each period. *Eqs. (7)* and *(8)* represent the sub-tour elimination constraints. The flow conservation constraints, which confirm the equality of the number of incoming and departing arcs at a vertex, are defined in *Eq. (9)*. *Eq. (10)* enforces that the amount of vehicle loading on a route should not exceed the amount assigned to that route's customers. Furthermore, *Eq. (11)* specifies that if the demand of customer i is not met in period t, no product will be sent to them. Finally, *Eq. (12)* guarantees that the vehicle's capacity is not exceeded. *Constraints (13)* and *(14)* further define restrictions for the variables.

## 4 | Genetic Algorithm

Biologically motivated approaches are particularly popular in solving complex optimization problems. The GA is a stochastic optimization approach constructed based on evolutionary processes inspired by the process of natural selection. Using operators such as crossover, mutation, and selection, the GA synthesizes

the good features of different individuals within the population in order to create individuals who are better suited. It is widely applied to solve different classes of NP-hard problems. The IRP belongs to the NP-hard class of problems, and exact solution methods are highly time-consuming for large-sized problems. Therefore, in the proposed problem, the GA approach is utilized. The flowchart of the general procedure of the proposed GA in this study is shown in *Fig. 1*. This procedure was adapted from the one presented by Gen et al. [34].

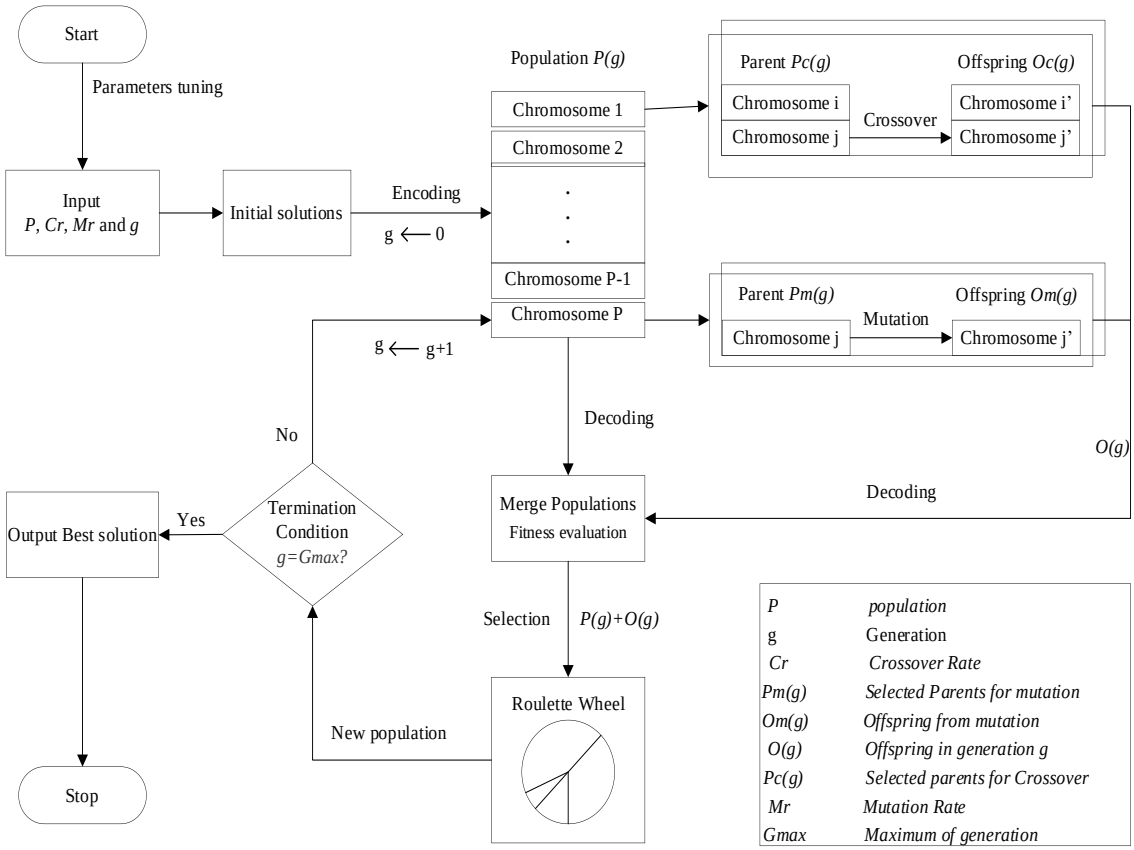


**Fig. 1. General procedure of the proposed GA.**

GA are initialized by a set of chromosomes (solutions) called the population. These chromosomes progress through successive iterations, known as generations. During each iteration, fitter chromosomes, evaluated by a fitness function, have higher chances of being selected to produce several offspring as new solutions. In the new population, which includes parents and children, the best individuals are chosen based on their fitness. The population size remains constant throughout all iterations. The GA may converge to the best solution after a certain number of iterations. The pseudo code for the GA proposed for the current problem is depicted in *Algorithm 1*.

Farahbakhsh and Kheirkhah| Int. J. Res. Ind. Eng. 12(4) (2023) 397-413

**Algorithm 1. The pseudo code for general procedure of the proposed GA.**

**Procedure:** GA for IRP
**Input:** Customers demand, Vehicle capacity, Planning horizon, Taguchi-based tuned parameters
**Output:** The best solution (vehicle route and amount of products assigned to customers)
**begin**
*g ¬ 0;*
initialize *P(g)* by encoding routine;
evaluate *P(g)* by decoding routine;
**while** (not termination condition) **do**
create $O_c(g)$ from *P(g)* by crossover routine;
create $O_m(g)$ from *P(g)* by mutation routine;
evaluate *O(g)* by decoding routine;
form *P(g+1)* out of *P(g)* and *O(g);*
set the best current solution;
*g* +1 ¬ *g*
**end**
**output:** The best solution
**end**

## 4.1 | Chromosome Representation

The chromosome representation and encoding of a solution is the first and most crucial task when utilizing a GA. Each chromosome must carry all the necessary information about the solution. In this problem, the chromosomes represent both served and unserved customers, as well as the route and amount of products transshipped to customers in each period.

The proposed encoded chromosome takes the form of a matrix with dimensions t *(c+k), where t, c, and k represent the number of time periods, number of customers, and number of vehicles available, respectively. Each row in the matrix is a permutation of real numbers between 1 and c+k, specifically:

- *Numbers 1 to c correspond to customers.*
- *Numbers c +1 to c +k represent vehicles.*

In the t'th row of the chromosome, genes related to customers appear before genes related to vehicles. These represent the customers served by the desired vehicle during period t. The vehicle route starts at the supplier and returns to the supplier after visiting the assigned customers. *Fig. 2* provides an example of the encoding procedure used in this paper.

**Vehicle usage**

If the first gene value in the chromosome string corresponds to a vehicle or if there are no genes related to customers between the two genes related to vehicles, that particular vehicle will not be used in this specific time period.

**Unserviced customers**

The gene values after the last vehicle gene up to the end of the chromosome string indicate those customers who were not serviced during this period.

**All customers visited**

Logically, if there are no other gene values after the last vehicle gene at the end of the chromosome string, it implies that all customers will be visited in this particular period.

This encoding ensures that each chromosome carries essential information about customer service, vehicle routes, and transshipment of products. It's a crucial step in utilizing a GA to solve the problem.



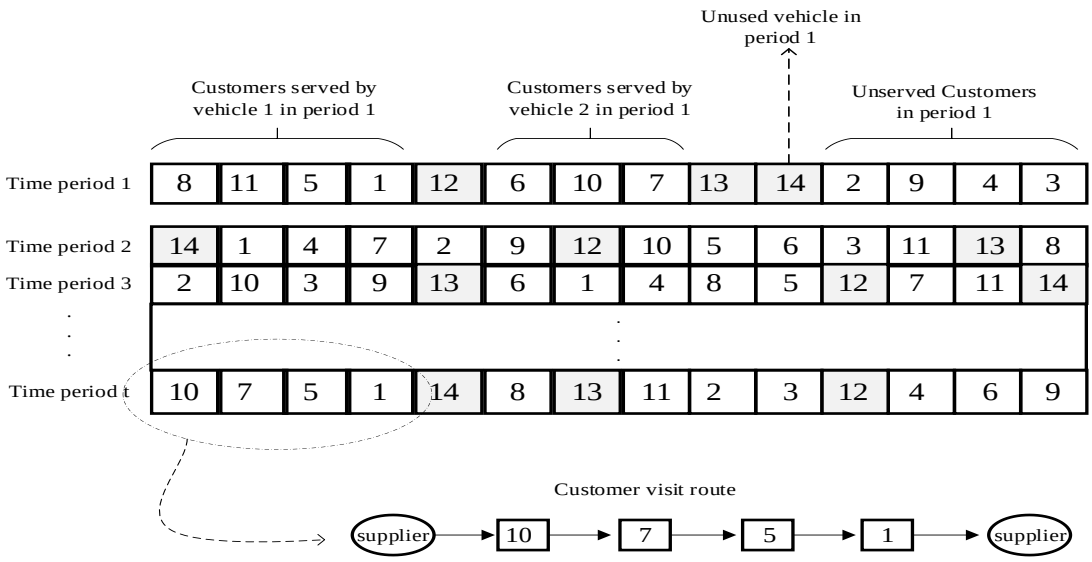


**Fig. 2. Chromosome representation.**

The example in *Fig. 2* refers to a chromosome with 11 customers and 3 vehicles, operating over t time periods. Gene values 1 to 11 correspond to customers, while genes 12 to 14 represent supplier-available vehicles. In time period 1, the value of genes from the beginning of the chromosome string up to the first gene related to vehicles (i.e., gene 12) represents customers serviced by the first vehicle. The genes after gene 12 up to the second gene related to vehicles (i.e., gene 13) are serviced by the second vehicle. This procedure is repeated for the number of available vehicles (i.e., 3). Customers assigned to each vehicle are served in the order of their appearance on the chromosome. This arrangement determines the route for each vehicle, which starts from the supplier and returns to the supplier after visiting the assigned customers (S-8-11-5-1-S).

## 4.2 | Initialization

To initiate the exploration of a nearly ideal solution, a population with the desired number of members is generated as an initial solution. Each individual chromosome in the population is represented by a t*(c+v) matrix, where t, c, and v denote the number of time periods, customers, and available vehicles, respectively. Each row of the matrix consists of a random permutation of real numbers between 1 and c+v.

## 4.3 | Chromosome Decoding

The values of the decision variable, $y_{ij}^{kt}$ are obtained from permutation numbers of the chromosome as follows (k regardless of its gene value, is the kth vehicle's gene in each row):

$$y_{ij}^{kt} = \begin{cases} 1, & \text{if in tth row, gene value j immediately follows gene value i} \\ & \text{in the distance between the kth and k−1th vehicles' gene,} \\ 0, & \text{Otherwise.} \end{cases} \tag{15}$$

The values of $qs_{ik}^{t}$ and $I_{it}$ for the customers according to the values of $y_{ij}^{kt}$ are then determined as follows:

$$qs_{ik}^{t} = \begin{cases} U_i - I_{i,t-1} + d_{i,t}, & \text{if } y_{ij}^{kt}=1, \quad \text{for all i,t,} \\ 0, & \text{Otherwise.} \end{cases} \tag{16}$$

$$I_{i,t} = \begin{cases} U_{i,} & \text{if } y_{ij}^{kt}=1, \quad \text{for all i,t,} \\ I_{i,t-1} - d_{i,t}, & \text{Otherwise.} \end{cases} \tag{17}$$

Farahbakhsh and Kheirkhah| Int. J. Res. Ind. Eng. 12(4) (2023) 397-413

Then, the values of $ql_{ij}^{kt}$ which is related to the amount of vehicle loading, are obtained by the chromosome depending on the values of $y_{ij}^{kt}$ and $qs_{ik}^{t}$ as follows:

$$ql_{ij}^{kt} = \sum_{i \in E} qs_{ik}^{t} \qquad \text{for all } j, k, t, \tag{18}$$

E: gene values' set include the gene value j and after that up to kth gene related to vehicles in row t.

## 4.4 | Evaluation

The fitness function is calculated according to the objective function to evaluate the solutions in the population. In this model, the fitness value is attained through the cost components of the objective function, along with two penalty terms. These cost components include the routing cost, the cost of using the vehicle (known as the fixed cost), and the inventory holding cost. According to the constraints of the model, penalties related to exceeding the capacity of the vehicle and unsatisfied customer demand in each period are also added to the value of the fitness function. After calculating the fitness function, genetic operators are applied to solutions with better-fitted values, and costly solutions are removed from the population.

## 4.5 | Genetic Operations

Genetic operators, which are generally categorized as selection, crossover, and mutation, are used to create better solutions and replace them with existing solutions.

### 4.5.1 | Selection

The chromosome chosen for genetic operations is determined by the roulette wheel operator. Each chromosome in the population is given a selection probability proportional to its fitness value. The fitter chromosomes have lower cost values and subsequently have higher selection probabilities. The selection probabilities are determined based on the total fitness value (F) according to *Eq. (19)*.

$$F = \sum_{h=1}^{popsize} fittness_h. \tag{19}$$

The selection probability Ph for each chromosome h is:

$$Ph = \frac{F - fittness_h}{F \times (popsize - 1)}. \tag{20}$$

Then, a random number r is generated in the range (0,1]. If $qh-1 < r \leq qh$, then chromosome h is selected.

### 4.5.2 | Crossover

Crossover, the primary GA operator, reproduces individuals by combining the data of parents who were chosen at random in such a way that the resulting offspring exhibit traits from both parents. The fitness function is used to compare these offspring and pass the information on to the following generation. One-point, two-point, and uniform crossover are three different types of crossover that can occur between chromosomes. Cyclic permutations are the best candidates for the two-point crossover. Another method for more quickly covering the search space is to reorganize the chromosomes after the crossover process. In this situation, the Order Crossover (OX) operator, which has proven effective in a variety of routing-related situations [33], is chosen to be used. This kind of crossover occurs as described below:

I. The first and last genes on the chromosome undergo a two-point crossover. Integer numbers are used to code the genes.
II. The two cut points' locations are chosen at random.
III. The genes between the cut points are first replicated in the offspring in the same order and location.
IV. The genes of the other parent are then copied in the same order, skipping the existing genes, starting from the second cut point of one of the parents.

The OX process to create the first offspring is demonstrated in the example in *Fig. 2*. The exchange of gene sequences between P1 and P2's cut points is what is done first. The first parent's genes from the second cut point are arranged in the following order: 5-7-6-1-3-4-2-8. After genes 7, 1, 4, and 2, which are duplicated in the first child, are deleted, a new sequence, 5-6-3-8, will be copied from the second cut point. The second child goes through the same procedure. The mentioned crossover operator is illustrated in *Fig. 3*.

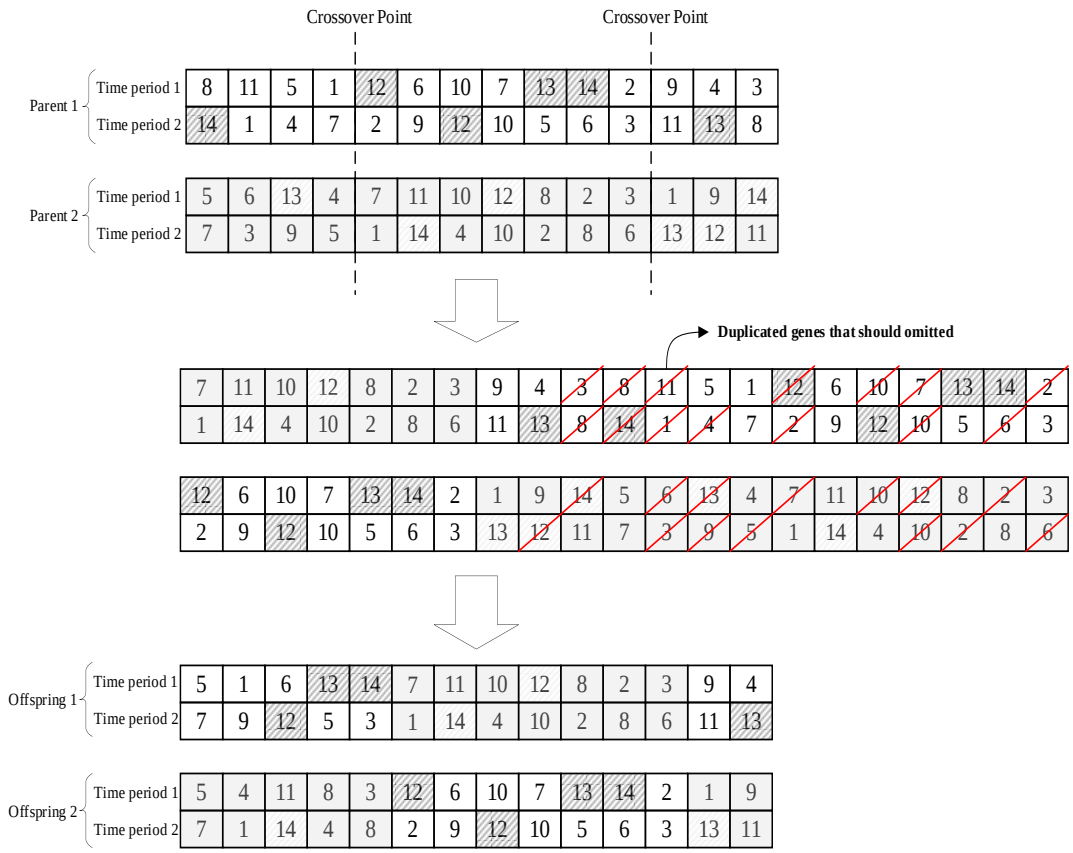


**Fig. 3. Order crossover OX process.**

### 4.5.3 | Mutation

Mutation is another genetic operator. The primary purpose of this operator is to investigate novel solutions in the solution space. Additionally, it is used to broaden the search space by randomly changing individual genes to avoid becoming trapped in local optima. In this study, three different mutations are used for the mutation process, as presented below:

**Swap mutation**

In swap mutation, two genes are selected randomly, and then their values are swapped in all time periods. Permutation is maintained, and perturbation is accomplished using swap mutation.

Farahbakhsh and Kheirkhah| Int. J. Res. Ind. Eng. 12(4) (2023) 397-413

**Reversion mutation**

In reversion mutation, a part of the chromosome is reversed. First, two genes are randomly selected, and then all the gene values between the two selected genes are reversed. This procedure is done for all time periods.

**Insertion mutation**

In the insertion mutation, two genes are randomly selected, and then the value of the first selected gene in all time periods is inserted after the second selected gene.

Each proposed mutation can explore new solutions in the solution space that other mutations are not able to explore. To select the mutation, the roulette wheel operation has been used, which selects mutations with unequal probabilities. The selection probability of mutations is determined according to the performance of the algorithm. The three mentioned mutations are shown in *Fig. 4*.

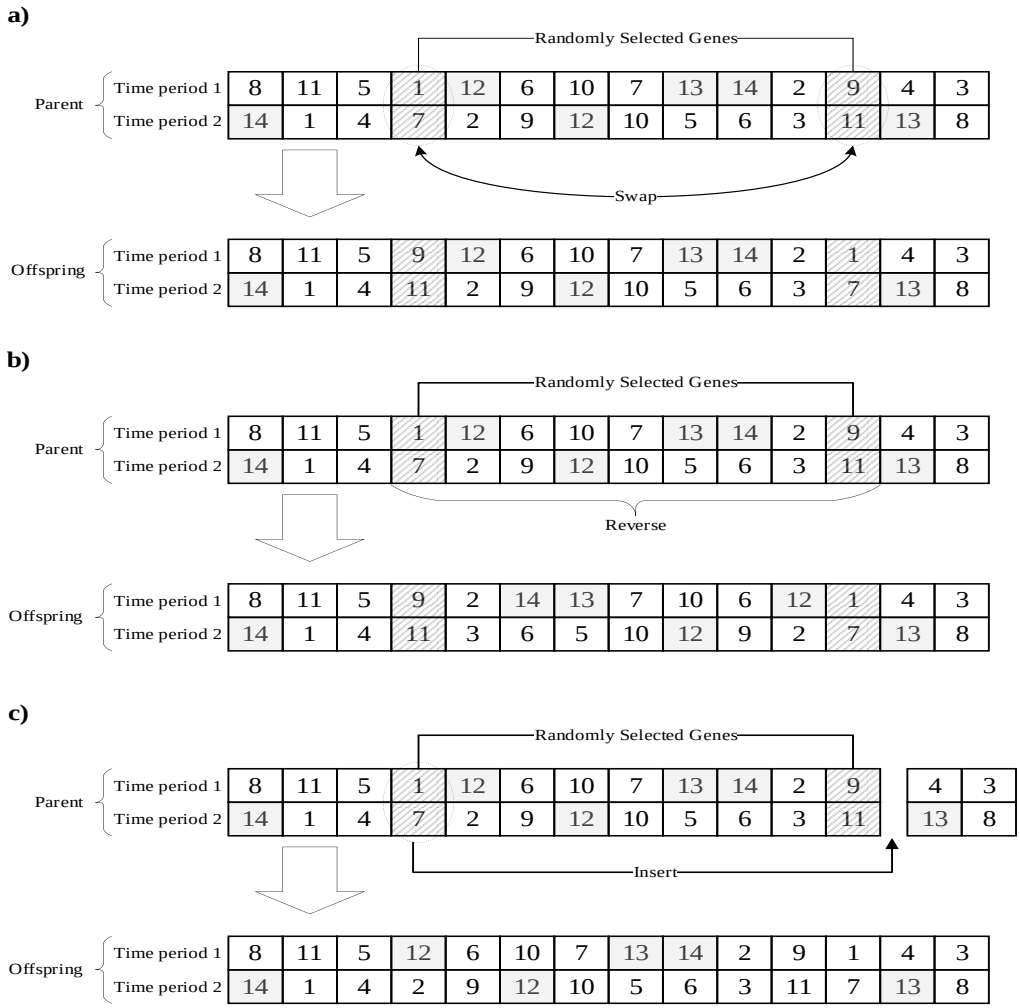


**Fig. 4. Mutation operation: a. Swap; b. Reversion; c. Insertion.**

### 4.5.4 | Stop condition

The algorithm stops when a predefined number of generations is reached.

## 5 | Taguchi-based Parameters Tuning

The appropriate configuration of parameters significantly impacts the effectiveness and efficiency of metaheuristic algorithms. Most research studies rely on either literature-based reference values or trial-and-

error to set these parameters. Since the ideal algorithmic parameters vary depending on the specific problem, newly developed algorithms require tailored adjustments to their parameters, resulting in improved algorithmic solutions. To estimate the appropriate algorithmic parameters, the Taguchi method is employed for optimizing the proposed GA's parameters. The Taguchi experimental design has found widespread application in optimization problems. It relies on two main tools: the Orthogonal Array (OA) and the Signal-to-Noise (S/N) ratio. The OA represents a numerical matrix containing experimental plans based on various levels of factors. The robustness of this experimental design is ensured by the S/N ratio, which quantifies variation. In this context, "signal" refers to the desired value (mean response variable), while "noise" corresponds to the undesirable value (standard deviation) [35]. 1) the Taguchi method aims to minimize the impact of noise while simultaneously determining optimal levels for controllable parameters based on robustness [36], and 2) for minimization problems, the goal is to maximize the S/N ratio for each parameter i at its level j, as calculated by *Eq. (21)*.

$$\left(\frac{S}{N\ ratio)_{ij}} = -10\log_{10}\left(\frac{\sum Z_{ij}^2}{n}\right)\right. \qquad \text{for all } i, j, \tag{21}$$

where n is the number of times level j of parameter i is repeated over the runs of all trials and Zij is the objective function value using parameter i on level j. *Algorithm 2* contains the proposed Taguchi method's pseudo code.

**Algorithm 2. The pseudo code for Taguchi-based tuning of the proposed GA.**

**Procedure:** Taguchi design for GA parameters
**Input:** Levels of *Iterarion-N, Population-size, Crossover-R, Mutation-R, Selection-P*
**Output:** Optimum level of parameters
**begin**
select $L_{16}(4^5)$ as the suitable OA;
apply GA on each sheme of $L_{16}(4^5)$;
obtain Z for each sheme;
**for** $i$ ¬ 1 to (5) **do**
**for** $j$ ¬ 1 to (4) **do**
calculate S/N ratio;
**end**
**end**
determine optimum level of parameters;
**end**

To implement the selected experimental design, the initial step in parameter configuration involves choosing the parameters that will act as controls and determining their corresponding levels. The proposed GA considers parameters such as population size, number of iterations, crossover rate, mutation rate, and selection pressure for fine-tuning. *Table 1* outlines the different levels in the tuning process for these parameters.

**Table 1. Different levels of GA parameters used for turning.**

| Factors (GA Parameters) | Levels | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| Population size | 80 | 100 | 120 | 140 |
| Number of iterations | 100 | 150 | 200 | 250 |
| Crossover rate | 0.7 | 0.75 | 0.8 | 0.85 |
| Mutation rate | 0.2 | 0.25 | 0.3 | 0.35 |
| Selection pressure | 6 | 7 | 8 | 9 |

L16 (45) was selected from the standard table of OAs. *Table 2* provides a summary of the sixteen different combinations of the constructed parameters and the outcomes attained through various parameter designs. In this table, the rows represent the parameter levels in each experimental scheme, and the columns represent the particular parameter levels that can be changed for each scheme.

Farahbakhsh and Kheirkhah| Int. J. Res. Ind. Eng. 12(4) (2023) 397-413

**Table 2. Results obtained from different designs of Taguchi approach.**

| Design | GA Parameters Population-Size | Iteration-N | Crossover-R | Mutation-R | Selection-P | Fitness Function |
|---|---|---|---|---|---|---|
| 1 | 80 | 100 | 0.7 | 0.2 | 6 | 5620.65 |
| 2 | 80 | 150 | 0.75 | 0.25 | 7 | 5201.23 |
| 3 | 80 | 200 | 0.8 | 0.3 | 8 | 5303.56 |
| 4 | 80 | 250 | 0.85 | 0.35 | 9 | 5905.84 |
| 5 | 100 | 100 | 0.75 | 0.3 | 9 | 5543.23 |
| 6 | 100 | 150 | 0.7 | 0.35 | 8 | 5757.52 |
| 7 | 100 | 200 | 0.85 | 0.2 | 7 | 5889.54 |
| 8 | 100 | 250 | 0.8 | 0.25 | 6 | 5856.69 |
| 9 | 120 | 100 | 0.8 | 0.35 | 7 | 5153.68 |
| 10 | 120 | 150 | 0.85 | 0.3 | 6 | 5914.68 |
| 11 | 120 | 200 | 0.7 | 0.25 | 9 | 5956.87 |
| 12 | 120 | 250 | 0.75 | 0.2 | 8 | 5380.56 |
| 13 | 140 | 100 | 0.85 | 0.25 | 8 | 5563.65 |
| 14 | 140 | 150 | 0.8 | 0.2 | 9 | 5152.84 |
| 15 | 140 | 200 | 0.75 | 0.35 | 6 | 5419.37 |
| 16 | 140 | 250 | 0.7 | 0.3 | 7 | 5682.75 |

Using Minitab 18, the proposed design was applied to each parameter at four levels. According to the mean for the answers and S/N ratio plot shown in *Fig. 5*, a good solution for the population size, number of iterations, mutation and crossover rates, and selection pressure is 120, 150, 0.3, 0.85, and 6, respectively.

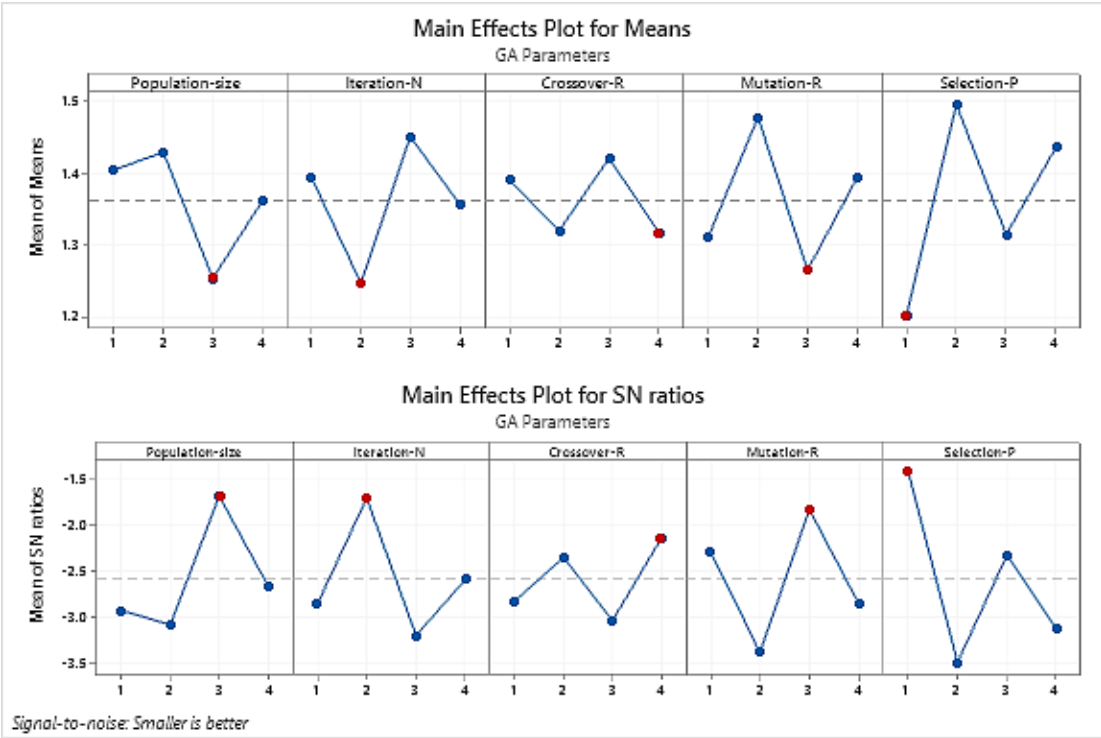


**Fig. 5. Comparison of the mean for the answers and S/N ratios of the algorithms.**

## 6 | Results Analysis

The performance of the developed GA was tested in this section. The modified GA has been implemented in MATLAB and run on a Core i5 and 8 GB RAM personal computer. The modified GA was tested on benchmark instances used in [37]. These instances are presented in two sizes: small with 5 to 50 customers and large with 50 to 200 customers. Each of the instances has been proposed over three and six time horizons. The OU policy was intended as a replenishment inventory policy, according to which every visit to a customer brings its inventory to the maximum level. In this research, relevant small instances have been used. The termination criterion is reaching the number of iterations, which is 150 iterations according to parameter tuning. Therefore, the number of iterations is fixed and running times are used for time

comparison. To achieve average results of executed instances, each instance was run 30 times. By making the problem more complex, the exact algorithm is less able to solve it in a reasonable time. *Table 3* displays results of developed metaheuristic as well as those from earlier studies that were applied to same benchmark.



**Table 3. Average result values obtained by the different resolution methods for the small-instances.**

| Instance | Optimal Solution | GA-OX | Time(s) | MA | Time(s) | Modified GA | Time(s) | ER |
|---|---|---|---|---|---|---|---|---|
| | Number of Time Periods = 3 | | | | | | | |
| Small-5 | 1418.76 | 1418.76 | 71.95 | 1418.76 | 83.14 | 1418.76 | 68.11 | 0 |
| Small-10 | 2228.67 | 2228.67 | 81.27 | 2228.67 | 98.49 | 2228.67 | 76.01 | 0 |
| Small-15 | 2493.47 | 2493.47 | 118.41 | 2493.47 | 179.31 | 2493.47 | 99.12 | 0 |
| Small-20 | 3053.02 | 3121.43 | 160.38 | 3053.02 | 221.17 | 3053.02 | 132.75 | 0 |
| Small-25 | 3451.15 | 3451.15 | 220.20 | 3451.15 | 238.53 | 3456.14 | 185.85 | 0.144589 |
| Small-30 | 3643.22 | 3643.22 | 232.84 | 3643.22 | 253.87 | 3643.22 | 201.31 | 0 |
| Small-35 | 3846.87 | 3958.73 | 248.13 | 3848.46 | 281.28 | 3848.46 | 213.44 | 0.041332 |
| Small-40 | 4125.70 | 4150.79 | 275.58 | 4136.57 | 296.43 | 4132.70 | 249.85 | 0.169668 |
| Small-45 | 4270.61 | 4279.19 | 293.91 | 4279.19 | 311.27 | 4279.19 | 286.53 | 0.200908 |
| Small-50 | 4810.87 | 4887.16 | 364.84 | 4811.92 | 409.31 | 4853.12 | 301.72 | 0.87822 |
| | Number of Time Periods = 6 | | | | | | | |
| Small-5 | 3299.98 | 3299.98 | 213.77 | 3299.98 | 249.23 | 3299.98 | 198.85 | 0 |
| Small-10 | 4832.89 | 4832.89 | 259.19 | 4832.89 | 301.87 | 4832.89 | 236.32 | 0 |
| Small-15 | 5566.39 | 5638.59 | 297.43 | 5566.39 | 366.30 | 5566.39 | 284.18 | 0 |
| Small-20 | 6833.29 | 6838.42 | 427.23 | 6833.29 | 528.54 | 6838.42 | 361.68 | 0.075074 |
| Small-25 | 7454.15 | 7475.88 | 483.59 | 7475.88 | 631.86 | 7462.11 | 416.79 | 0.106786 |
| Small-30 | 7847.39 | 7899.12 | 713.74 | 7868.36 | 843.39 | 7877.52 | 506.60 | 0.383949 |

The size of the instance sets is shown in the first column. The second column shows the optimal solution to the problem solved in [37]. The third, fifth, and seventh columns show the average of the best outcomes obtained with GA-OX and MA of [33] and modified GA developed in this paper, respectively. The average computation time needed to obtain the results from GA-OX, MA, and modified GA is shown in the fourth, sixth, and eighth columns, respectively. The Error Rate (ER) of the modified GA, in the ninth column, was computed with *Eq. (22)*.

$$\mathrm{ER} = 100 * \left( \frac{\text{total cost obtained using the modified GA} \; - \; \text{optimal solution}}{\text{optimal solution}} \right). \tag{22}$$

As can be seen, the developed algorithm shows better performance in solving instances. The calculated ER is 0% in many cases and up to 0.87% in other cases. The ER in these instances does not even reach 1%, which indicates good performance of the algorithm in the problem due to low time to solve it. Modified GA was able to accurately solve five instance sets of three delivery periods and four instance sets of six delivery periods. By examining the percentage of solution improvement in these instances, it shows the efficiency of the modified GA algorithm. Two algorithms, MA and GA-OX, have cumulative deviations of 1.09% and 9.86% from optimal solution in all instances, respectively. The deviation percentage of proposed algorithm is 1.14%, which is very good compared to GA-OX algorithm. Also, compared to MA algorithm, although difference is very small considering solving time, better performance is obtained from presented algorithm. Better performance of proposed algorithm compared to other two algorithms can be seen not only in value of objective function but also in time taken to reach solution of problem. Modified GA has achieved optimal or close to optimal solution in less time than MA and GA-OX in all instances. This issue can be seen in the diagram in *Fig. 6*. The effect of reducing the solution time will be more pronounced in larger samples.

Farahbakhsh and Kheirkhah| Int. J. Res. Ind. Eng. 12(4) (2023) 397-413

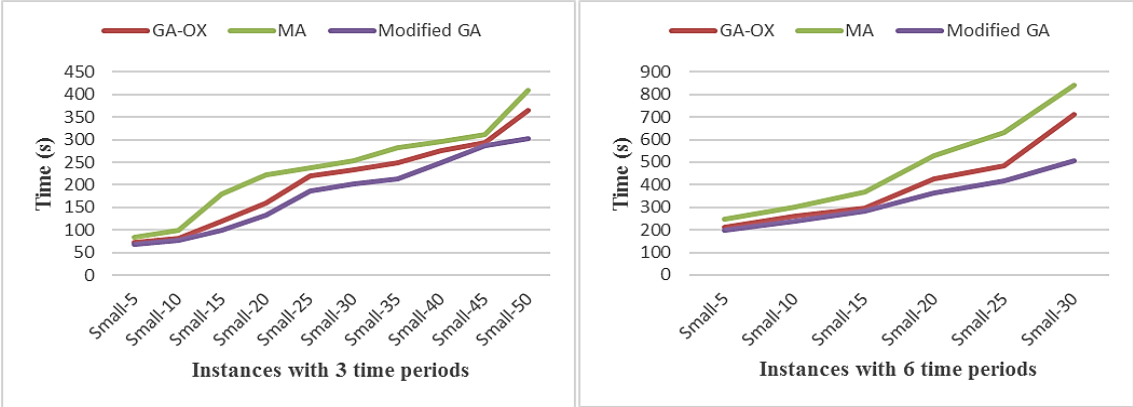


**Fig. 6. Comparison of solving times of the algorithms.**

Premature convergence in the local optimum is one of the problems of the GA that prevents reaching the global optimum. *Fig. 7* shows the convergence behavior of the proposed GA over 150 generations. This diagram is related to the solution of the first standard example. As it is clear in the figure, the algorithm has converged to local solutions in generations 22 to 30 and 36 to 50, but due to the defined mutation, the algorithm has been able to exit this convergence and continue the solution procedure. The solution converges to the global optimum at 61 generations, which is relatively fast.

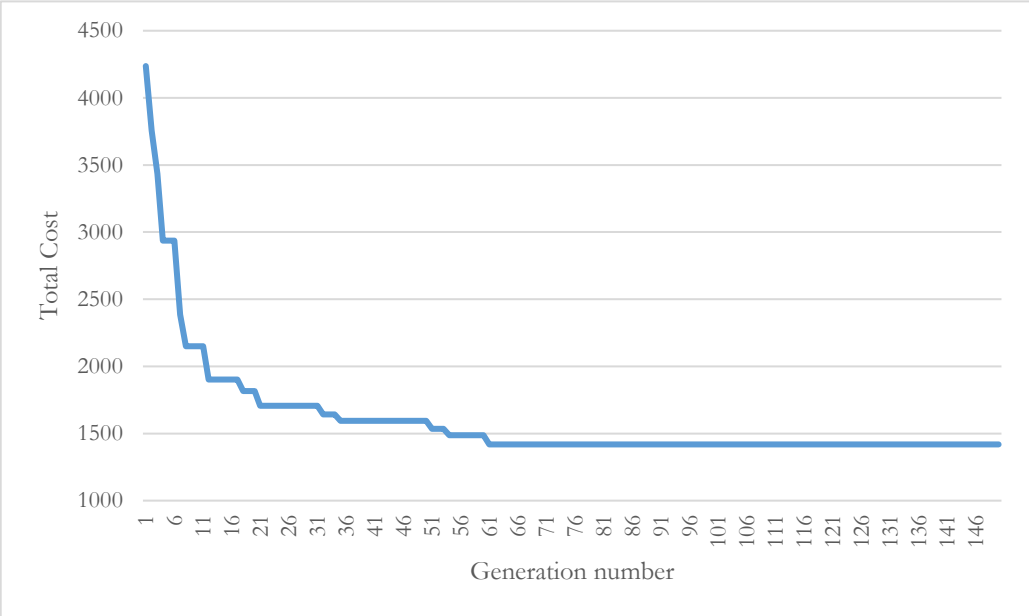


**Fig. 7. Convergence behavior of the proposed GA over 150 generations.**

## 7 | Conclusions

The IRP has gained much attention from practitioners in the literature. The problem's complexities imply more use of meta-heuristic algorithms to solve it. The GA is widely used to solve optimization problems. Various approaches have been employed for the chromosomal structure of the GA based on the under-investigation problem, all of which try to improve the performance of the problem algorithm. Introducing a novel chromosomal structure based on IRP, this study could use GA with better conditions. Simple chromosome structure, encrypting and decrypting the chromosome, and higher speed of the algorithm are some characteristics of this structure. To evaluate and investigate the algorithm performance, IRP has been modeled as a linear single-objective mixed integer programming. The parameters of the algorithm are adjusted and determined by using Taguchi method with 16 trials. To compare algorithm efficiency, 26 standard samples with small and large sizes available in the literature were used. Due to the random nature of the algorithm, each sample was solved 30 times, and the average values were used to perform comparisons. A comparison of the results with those in other studies shows that modified algorithm has

good performance. In many cases, this algorithm provided higher-quality solutions than its counterparts. In all samples compared, presented algorithm performed better concerning time and could reach solutions in a shorter time. As a suggestion for the future, one can develop the same chromosomal structure for PRP. As well, this algorithm can be combined with others to improve performance.



## References

[1] 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. DOI:10.1016/j.ijpe.2010.06.012

[2] Bertazzi, L., Coelho, L. C., De Maio, A., & Laganà, D. (2019). A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation research part e: logistics and transportation review*, *122*, 524–544. https://doi.org/10.1016/j.tre.2019.01.005

[3] Coelho, L. C., Cordeau, J. F., & Laporte, G. (2014). Thirty years of inventory routing. *Transportation science*, *48*(1), 1–19. DOI:10.1287/trsc.2013.0472

[4] Mahjoob, M., Fazeli, S. S., Milanlouei, S., Tavassoli, L. S., & Mirmozaffari, M. (2022). A modified adaptive genetic algorithm for multi-product multi-period inventory routing problem. *Sustainable operations and computers*, *3*, 1–9. DOI:10.1016/j.susoc.2021.08.002

[5] Archetti, C., Feillet, D., & Speranza, M. G. (2015). Complexity of routing problems with release dates. *European journal of operational research*, *247*(3), 797–803. https://doi.org/10.1016/j.ejor.2015.06.057

[6] Lambora, A., Gupta, K., & Chopra, K. (2019). Genetic algorithm-a literature review. *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCON)* (pp. 380–384). IEEE. https://ieeexplore.ieee.org/abstract/document/8862255

[7] Mirjalili, S. (2019). Evolutionary algorithms and neural networks. In *Studies in computational intelligence* (Vol. 780). Springer. https://doi.org/10.1007/978-3-319-93025-1

[8] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia tools and applications*, *80*, 8091–8126. https://link.springer.com/article/10.1007/s11042-020-10139-6

[9] Zapata-Cortes, J. A., Arango-Serna, M. D., Serna-Úran, C. A., & Gil-Gómez, H. (2021). A genetic algorithm for solving the inventory routing problem with time windows. In *New perspectives on enterprise decision-making applying artificial intelligence techniques* (pp. 463–481). Springer. https://doi.org/10.1007/978-3-030-71115-3_20

[10] Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., … & Prutzman, P. J. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces (Providence, Rhode Island)*, *13*(6), 4–23. DOI:10.1287/inte.13.6.4

[11] Cheng, C., Qi, M., Wang, X., & Zhang, Y. (2016). Multi-period inventory routing problem under carbon emission regulations. *International journal of production economics*, *182*, 263–275. DOI:10.1016/j.ijpe.2016.09.001

[12] Onggo, B. S., Panadero, J., Corlu, C. G., & Juan, A. A. (2019). Agri-food supply chains with stochastic demands: a multi-period inventory routing problem with perishable products. *Simulation modelling practice and theory*, *97*, 101970. DOI:10.1016/j.simpat.2019.101970

[13] Xiao, N., & Rao, Y. L. (2016). Multi-product multi-period inventory routing optimization with time window constrains. *International journal of simulation modelling*, *15*(2), 352–364. DOI:10.2507/IJSIMM15(2)CO8

[14] Alinaghian, M., Tirkolaee, E. B., Dezaki, Z. K., Hejazi, S. R., & Ding, W. (2021). An augmented Tabu search algorithm for the green inventory-routing problem with time windows. *Swarm and evolutionary computation*, *60*, 100802. DOI:10.1016/j.swevo.2020.100802

[15] Soysal, M., Çimen, M., Belbağ, S., & Toğrul, E. (2019). A review on sustainable inventory routing. *Computers & industrial engineering*, *132*, 395–411. https://doi.org/10.1016/j.cie.2019.04.026

[16] Solyalı, O., & Süral, H. (2011). A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation science*, *45*(3), 335–345. DOI:10.1287/trsc.1100.0354

[17] Coelho, L. C., & Laporte, G. (2014). Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers and operations research*, *47*, 42–52. DOI:10.1016/j.cor.2014.01.013

[18] Andersson, H., Christiansen, M., & Desaulniers, G. (2016). A new decomposition algorithm for a liquefied natural gas inventory routing problem. *International journal of production research*, *54*(2), 564–578. DOI:10.1080/00207543.2015.1037024

[19] Oudouar, F., & Zaoui, E. M. (2019). A hybrid Heuristic for vehicle routing problem. *ACM international conference proceeding series*, *24*(1), 101–116. DOI:10.1145/3372938.3373014

[20] Yu, B., Ma, N., Cai, W., Li, T., Yuan, X., & Yao, B. (2013). Improved ant colony optimisation for the dynamic multi-depot vehicle routing problem. *International journal of logistics research and applications*, *16*(2), 144–157. DOI:10.1080/13675567.2013.810712

[21] Cordeau, J. F., Laganà, D., Musmanno, R., & Vocaturo, F. (2015). A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & operations research*, *55*, 153-166. https://doi.org/10.1016/j.cor.2014.06.007

[22] Su, Z., Lü, Z., Wang, Z., Qi, Y., & Benlic, U. (2020). A matheuristic algorithm for the inventory routing problem. *Transportation science*, *54*(2), 330–354. DOI:10.1287/trsc.2019.0930

[23] Wu, W., Zhou, W., Lin, Y., Xie, Y., & Jin, W. (2021). A hybrid metaheuristic algorithm for location inventory routing problem with time windows and fuel consumption. *Expert systems with applications*, *166*, 114034. DOI:10.1016/j.eswa.2020.114034

[24] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

[25] 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. DOI:10.1080/00207540600597138

[26] Park, Y. B., Yoo, J. S., & Park, H. S. (2016). A genetic algorithm for the vendor-managed inventory routing problem with lost sales. *Expert systems with applications*, *53*, 149–159. DOI:10.1016/j.eswa.2016.01.041

[27] Othman, S. N., Mustaffa, N. H., Radzi, N. H. M., Sallehuddin, R., & Bazin, N. E. N. (2016). A modelling of genetic algorithm for inventory routing problem simulation optimisation. *International journal of supply chain management*, *5*(4), 43–51.

[28] Hiassat, A., Diabat, A., & Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of manufacturing systems*, *42*, 93–103. DOI:10.1016/j.jmsy.2016.10.004

[29] Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers and industrial engineering*, *104*, 124–133. DOI:10.1016/j.cie.2016.12.019

[30] Arab, R., Ghaderi, S. F., & Tavakkoli-Moghaddam, R. (2020). Bi-objective inventory routing problem with backhauls under transportation risks: two meta-heuristics. *Transportation letters*, *12*(2), 113–129. DOI:10.1080/19427867.2018.1533624

[31] Yavari, M., Enjavi, H., & Geraeli, M. (2020). Demand management to cope with routes disruptions in location-inventory-routing problem for perishable products. *Research in transportation business and management*, *37*, 100552. DOI:10.1016/j.rtbm.2020.100552

[32] Amri Sakhri, M. S. (2022). Comparative analysis of different crossover structures for solving a periodic inventory routing problem. *International journal of data science and analytics*, *14*(2), 141–153.

[33] Sakhri, M. S. A., Tlili, M., & Korbaa, O. (2022). A memetic algorithm for the inventory routing problem. *Journal of heuristics*, *28*(3), 351–375. DOI:10.1007/s10732-022-09497-1

[34] Gen, M., Cheng, R., & Lin, L. (2008). *Network models and optimization: multiobjective genetic algorithm approach*. Springer Science & Business Media.

[35] Freddi, A., Salmon, M., Freddi, A., & Salmon, M. (2019). Introduction to the Taguchi method. *Design principles and methodologies: from conceptualization to first prototyping with examples and case studies*, 159–180. https://doi.org/10.1007/978-3-319-95342-7_7

[36] Karna, S. K., & Sahai, R. (2012). An overview on Taguchi method. *International journal of engineering and mathematical sciences*, *1*(1), 1-18.

[37] Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation science*, *41*(3), 382–391.