



A Comparison between Memetic Algorithm and Genetic Algorithm for an Integrated Logistics Network with Flexible Delivery Path

Elham Behmanesh¹ · Jürgen Pannek²

Received: 28 October 2020 / Accepted: 19 July 2021 / Published online: 15 September 2021
© The Author(s) 2021

Abstract

The distribution/allocation problem is known as one of the most comprehensive strategic decision. In real-world cases, it is impossible to solve a distribution/allocation problem in traditional ways with acceptable time. Hence researchers develop efficient non-traditional techniques for the large-term operation of the whole supply chain. These techniques provide near optimal solutions particularly for large-scale test problems. This paper presents an integrated supply chain model which is flexible in the delivery path. As the solution methodology, we apply a memetic algorithm with a neighborhood search mechanism and novelty in population presentation method called “extended random path direct encoding method.” To illustrate the performance of the proposed memetic algorithm, LINGO optimization software serves as comparison basis for small size problems. In large-size cases that we are dealing with in real world, a classical genetic algorithm as the second metaheuristic algorithm is considered to compare the results and show the efficiency of the memetic algorithm.

Keywords Integrated logistics network · Flexible path · Memetic algorithm · Genetic algorithm

1 Introduction

The distribution/allocation problem is considered as an important strategic decision. If the locations of all central facilities are known in advance, but the assignment of flows between the set of central facilities and some other set of points are unknown, then

✉ Elham Behmanesh
beh@biba.uni-bremen.de; elham.bhsh@gmail.com

¹ International Graduate School for Dynamics in Logistics, University of Bremen, Bremen, Germany

² BIBA Bremer Institut für Produktion und Logistik GmbH, University of Bremen, Hochschulring 20, Bremen 28359, Germany

the problem becomes a distribution-allocation problem. If the assignment of flows is known, but the geographical disposition of the central facilities is unknown, then the problem is a pure location-allocation problem. The goal of location-allocation is to locate facilities in a way that supplies demand points most efficiently. As the name suggests, location-allocation is a twofold problem that simultaneously locates facilities and allocates demand points to the facilities.

In many countries, according to the environmental laws, companies are forced to equip their facilities in order to make the recovery of used products possible. As an example, In the last decade governmental legislation forced companies to collect, recover and recycle, e.g., all electronic devices in Europe, Japan, China, and many parts of USA and Canada as well as safe disposal of their goods [1]. The importance of this issue reveals some countries to ratify an annual recovery rate [2]. Also there are some pick-up policy for logistics network problems that make this procedure different for any country [2]. Except environmental damages happened by ignoring reverse flow, shortage in resources as the second problem is appeared. As a consequence, closed-loop supply chain has become a pressing topic for supply chain partners.

Environmental factors are not the only reason that spurs researchers or manufacturers to consider closed-loop logistic networks. In fact, the reason that makes recovery system attractive for companies is not environmental factors. The reverse activities represent an economic added value that can be obtained by processing the returned products and capturing the remaining value in the used products. There are some research prove the economic impact of adding reverse flow by numbers. One of them was focused on the gold obtained from electronic computer waste. The results demonstrate one ton of electronic computer waste can produce more gold than 17 tons of material extracted from a gold mine [3]. In another research, Xerox Europe company considered as a case study and the final results show the take back used products strategy resurrect 80 million dollars for the company [4].

By considering reverse distribution, some facilities need to be established such as collection/inspection center to collect and inspect the products under expert supervision as well as disposal center to have a safe disposal for non-recyclable items. Also some sections in plants need to be updated to be able for recycling or recovery of used products. Therefore, job creation can be considered as the other aspect of a closed-loop supply chain network.

According to the above explanation, it is realized that reverse distribution is important and need to be investigated in each supply chain network. Most researchers in this area, focused on only reverse flows in the network while considering the reverse and forward flow at the same time can reveal a better results [5]. Figure 1 illustrates the considered extended supply chain network. In this study, the reverse flow including collection/inspection center as well as disposal center added to the traditional network. In the backward flow, used products are collected by collection center and after inspection, sorting, and disassembly, the recoverable products are shipped to the plant for further operation and return to the cycle while scrapped products are transferred to disposal center for a safe disposal. It is important to note that remanufacturing process can occur in a separated facility or in plants where new products are produced. The same idea is considerable for collection and inspection

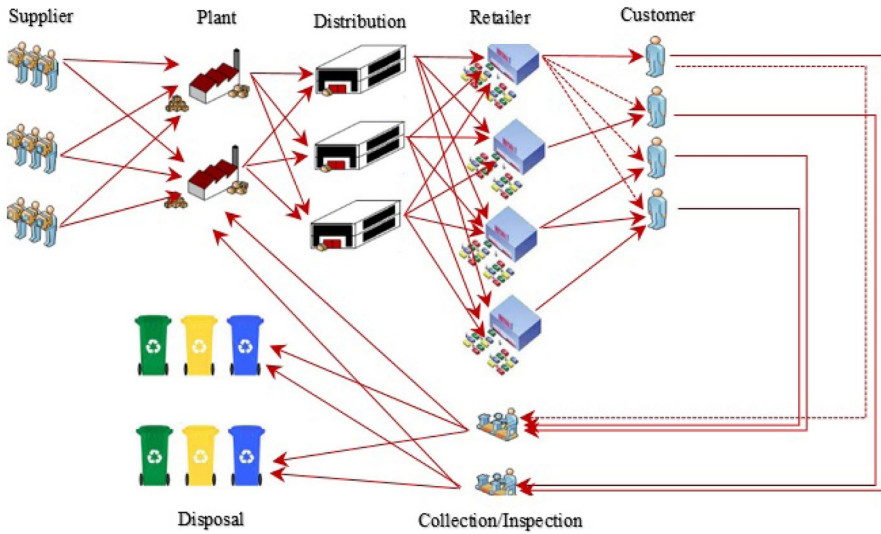


Fig. 1 Extended supply chain network

center. It is clear that establishing several facilities at the same location can reduce the cost of a supply chain network, in comparison with separated design.

Customer satisfaction creates a positive effect on any organization’s profitability. Satisfied customers form the foundation of any successful business as customer satisfaction leads to repeat purchase, brand loyalty, and positive word of mouth. Fast and on time delivery of products with a great accuracy plays an important role in customer satisfaction. The ability of supply chain to satisfy the customer’s expected delivery time is called “supply chain responsiveness” [6, 7]. On the other hand, existing an alternative shortest path may reduce transportation cost and accordingly the total cost of network.

Some studies optimized the supply chain network by considering both aspects, cost efficiency and responsiveness [6, 8]. However, most of these researches limited themselves by considering flow between facilities which are consecutive [6, 8, 9]. Nevertheless, having flow between facilities which are not consecutive can improve the efficiency of supply chain network. This configuration allow us to skip some stages and go for shortest path. Although considering flexibility in delivery path can increase efficiency of supply chain network, make the problem to be more complex, cf. Figure 2 for a sketch of the possible configuration by applying flexible logistics network. By considering additional delivery paths, we provide a full capacitated graph between plant to customer.

According to the above description, within this study, we add reverse distribution through an integrated design to the presented network. This avoids suboptimal solution revealed by separated design [5] and increases efficiency of the network by sharing number of resources. Also, except normal delivery which is defined from any stage to the other one, two additional delivery paths named, direct delivery and direct shipment are added to the network to increase the flexibility and performance

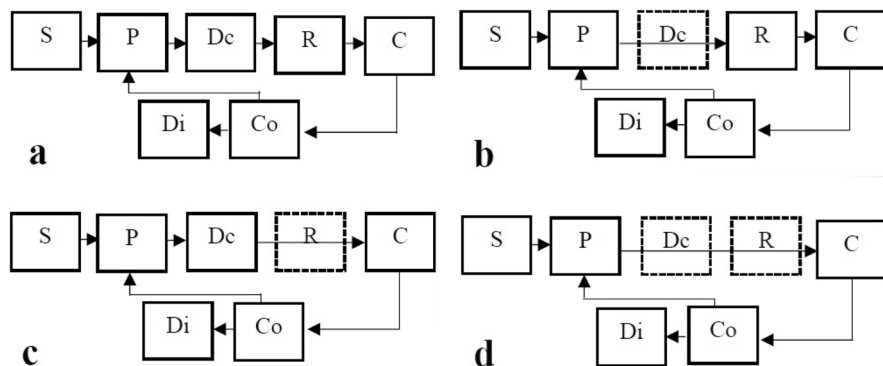


Fig. 2 Possible configuration for the proposed flexible network

of the supply chain network. Direct delivery, connects plants to retailers and also distribution centers to customers by skipping distribution centers and retailers respectively. Products can ship from plants to customers through direct shipment as well. Therefore, in addition we are facing with an integrated forward/reverse logistic network with a full delivery graph between plant to customer. This network is defined as a single period, single product and multi stage network. The model presents an NP-hard problem that is not possible to solve through traditional methods in acceptable time, particularly in large size. In this regard, many researchers have been developed heuristics and metaheuristics to cope with this NP-hard [6, 9–14]. Although as the network presented in this study is not a case-based network, developing an efficient solution methodology to tackle this NP hard problem is still a critical need in this area [15].

With in this study, we consider a memetic algorithm with a novelty in chromosome representation called “extended random path direct encoding method” and local search engine to enhance its search ability for the proposed flexible model and optimize the design of the supply chain network [16]. The reasons that we apply the extended random path direct encoding method can be summarized as follow:

- To deal with the complexity of different delivery paths.
- To integrate integrated logistics problem into memetic algorithms, using direct encoding.
- To reduce size of encoding.
- To decrease computational time.

Our aim is to show the efficiency of the proposed algorithm compared to a classical genetic algorithm particularly in large-size problems. The rest of this paper is organized as follows: The literature review and problems definition are introduced in Section 2. In Section 3, the proposed memetic algorithm as the main solution methodology is discussed briefly. Section 4 give the problem setting and computational results and finally, conclusion and suggestion for future work are presented in Section 5.

2 Literature Review and Problem Definition

There are some studies regarding the application of memetic algorithm and comparison between different metaheuristic algorithms. It should be noted that, many researchers applied some algorithms close to memetic algorithm but named them differently. Other names such as hybrid genetic algorithm, modified genetic algorithm, adapted genetic algorithm or Genetic local search algorithm in the literature can be considered as a memetic algorithm.

Pishvae et al. [6] proposed an integrated design for a logistic network to avoid suboptimality may obtain from separated design. Although the model can be formulated into a biobjective mixed integer linear programming, traditional methods are not capable to find a solution in acceptable time. The aim of this study is minimizing the total cost and maximize the responsiveness of the whole network. They applied a multi-objective memetic algorithm as the solution methodology to find the set of non-dominated solution. A new dynamic search strategy using three different local searches is proposed in this study. To verify the performance of the memetic algorithm, the pareto optimal solution obtained by the proposed MA is compared by a multi-objective genetic algorithm and LINGO optimization software. The numerical results obtained form error percent proved that the proposed memetic algorithm outperformed than the considered genetic algorithm.

Finding an optimal design of supply chain is discussed by Yun et al. [17] for a multistage network. In this study a hybrid genetic algorithm (hGA) with an adaptive local search engine based on hill climbing method is applied. Two measures of performance is considered in this research. First is focused on the average CUT time and second the time that all tasks of the longest route are completely finished. To show the efficiency of the proposed hGA, the proposed algorithm is compared to four different algorithms such as enumeration method, GA and another hGA with a local search technique and without a local search scheme. Using numerical results, they proved the proposed hGA works better than other mentioned algorithms.

An efficient memetic algorithm combined with a local search method presented by Moghaddam et al. [18] to solve a flexible flow line scheduling consist of processor blocking. In this study a new chromosome representation and operators are proposed for the solution methodology . The efficiency of the proposed memetic algorithm is versified based on a comparison to a classical genetic algorithm under the same condition through numerical results. The obtained results showed that the classical GA can not be considered as an efficient method for this kind of problem while the proposed memetic algorithm showed a good efficiency and capability. A comparison regarding convergence rate between the memetic and genetic algorithm is adapted as well. The results reveal a very sharp convergence of the MA compared by the GA.

Lin et al. [19] presented a memetic algorithm for a path planning and formation control of swarm robots. The goal was obtaining the best configuration of robot swarm when the evolution reaches a stability state. A non-random initial population is applied for the proposed algorithm. They compared the memetic

algorithm to a traditional genetic algorithm through numerical results. They observed the memetic algorithm works better in comparison with the traditional genetic algorithm due to the local search scheme for searching optimal solution. Also they showed the high efficiency of the proposed algorithm for large swarms.

A mixed integer linear programming is introduced by Lee and Dong [5]. The model considered an integrated design of forward and reverse network for end-of-lease computer products. As the solution methodology the authors presented a Tabu search-based memetic algorithm.

Another integrated forward/reverse logistics network is developed by Ko and Evans [20]. They formulated the model into a mixed integer nonlinear programming. They introduced a 3-PLS service providers to design the proposed network. A genetic algorithms is considered to cope with complexity of the proposed NP hard problem.

Wang et al. [21] established profit distribution plans using the improved Shapley value model. Optimal sequential coalitions were selected based on a strictly monotonic path, cost reduction model, and best strategy of sequential coalition selection in cooperative game theory. Following an empirical study in Chongqing, China, they suggested that the proposed approach outperforms other algorithms, and the best sequential coalition can be selected and adjusted to increase the negotiation power for network optimization of logistics distribution.

Wang et al. [22] optimized a location-routing problem by using eco-packages, in a state-space-time network. Their optimization involves solving a two-echelon location routing problem and the pickup and delivery problem with time windows. They showed that the 3D sharing state-space-time (SST) network representation in the first echelon annotatively captured the eco-package route sequence and state transition constraints over the shortest path in the pickup and delivery at any given moment of the transport phase.

In another study, Wang et al. [23] proposed and presented a collaborative two-echelon multicenter vehicle routing problem based on a state-space-time (CTMCVRP-SST) network to facilitate collaboration and resource sharing in a multiperiod state-space-time (SST) logistics network. A three-component solution framework is proposed to solve CTMCVRP-SST: a) A biobjective linear programming model, b) an integrated algorithm consisting of SST-based dynamic programming (DP), improved K-means clustering and improved non-dominated sorting genetic algorithm-II, c) a cost gap allocation model. Results showed that the proposed algorithm outperforms existing algorithms in minimizing the total cost with all other constraints being the same.

Fu et al. [24] proposed memetic algorithm MA-TOSCA to help WSNs resist cascading failures via topology optimization, where they designed the local search operator based on a new network balancing metric “sink-oriented betweenness entropy”. Following their simulations, they have shown that the proposed model can properly characterize the cascading process of WSNs and MA-TOSCA can find more robust topology with less time compared to existing algorithms.

Fu et al. [25] developed a new opportunistic network framework WAON (WSN-Assisted Opportunistic Network). They proposed a forwarding mechanism NetSpray for WAON, which supports four forwarding operations (mobile-to-mobile, static-to-mobile, mobile-to-static, and static-to-static). Their simulation and practical experiments showed that NetSpray can fully exploit the message-synchronization capability

of WAON, and had better performance than existing forwarding mechanisms in disaster scenarios.

In another study Fu et al. [26] designed a routing-based cascading model of WSNs and presented a routing recovery mechanism. In the proposed model, the load of a sensor node was defined as the number of data packets was limited by its congestion state. Extensive simulations have shown that the cascading invulnerability of WSNs is positively related to the overload-tolerance coefficient and inversely related to the congestion-tolerance coefficient, and the time required for overloaded nodes to reboot should be as short as possible.

Grac [27] compared memetic algorithm and genetic algorithm in order to investigate the performance for the cryptanalysis on simplified data encryption standard problems(SDES). The various methods and experiments were studied, and it was concluded that the memetic algorithm performed better than the genetic algorithms for such type of NP-Hard combinatorial problem.

Yadegari et al. [28] developed a memetic algorithm with an extended priority-based encoding/decoding method based on a flexible combinatorial neighborhood search mechanism. As discrete solution representation applied in this study is time consuming, they applied a technique to convert the discrete representation to a continuous one to deal with. Moreover a multi-start simulation annealing is considered to speed up the proposed algorithm.

Now we describe the proposed supply chain network mathematically. We assume the number of possible open facilities including suppliers, plants, distribution centers, retailers, customers, collection/inspection centers and disposal centers as well as their maximum capacities are known. The requested demand, returned rate p_j^{return} and disposal rate $(1 - p_j^{\text{disposal}})$ are also known. The model is aim to minimize the total cost including transportation cost and operation cost while the subset of facilities to be opened are chosen to design the optimum configuration of the supply chain and satisfy demand requested by customers. In this regard the problem is formulated by applying the following mixed integer linear programming (MILP) model while we consider a connected graph $G = (V, E)$ where $V = \{1, 2, \dots, n\}$ is a finite set of nodes and $E = \{(i, j) | i, j \in V\}$ is a finite set of edges which represent connection between the nodes.

$$\begin{aligned}
 \min_{x_{ij}, y_i} \quad & \sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{i \in V} c_i y_i \\
 \text{s.t.} \quad & \sum_{(i,j) \in E} a_i x_{ij} \leq b_i y_i \\
 & x_{ij} \geq 0, y_i \in \{0, 1\}
 \end{aligned} \tag{1}$$

Each node $i \in V$ has an associated number denotes by c_i representing the fixed cost of considered nodes. Also each edge $(i, j) \in E$ is connected with a number, denotes by c_{ij} , defines the unit transportation cost between the interested nodes. Two decision variables $y_i \in \{0, 1\}$ and $x_{ij} \in \mathbb{N}_0$ represent whether a stage $i \in V$ is used and which quantity is shipped between node i and j . Some conditions are considered and presented in [16] to adapt problem 1. We would like to note that the set of nodes

V in this study are given by the set of suppliers S , plants P , distribution centers Dc , retailers R , customers C , collection/inspection centers Co and disposal centers Di . The constraints for the presented model can be categorized into three parts.

1. capacity constraints in each node:

$$\sum_{(i,j) \in E} x_{ij} \leq \begin{cases} b_i & \forall i \in S \\ b_i y_i & \forall i \in V \setminus \{S \cup C\}. \end{cases} \quad (2)$$

2. The in-flow and out-flow in each node which must be identical.

$$\sum_{(j,k) \in E} x_{jk} = \begin{cases} \sum_{(i,j) \in E} x_{ij} & \forall j \in V \setminus \{C \cup Co\} \\ p_j^{\text{return}} \sum_{(i,j) \in E} x_{ij} & \forall j \in C \\ p_j^{\text{disposal}} \sum_{(i,j) \in Co \times Di} x_{ij} & \forall j \in Co \\ (1 - p_j^{\text{disposal}}) \sum_{(i,j) \in Co \times P} x_{ij} & \forall j \in Co \end{cases} \quad (3)$$

3. The demand of customers which must be satisfied.

$$\sum_{(i,j) \in E} x_{ij} = b_j \quad \forall j \in C \quad (4)$$

3 Solution Approach

Recently, evolutionary algorithms (EAs) have been successfully applied to solve hard optimization problems [29–33]. EAs are population-based global search methods based on the mechanism of natural selection in biological systems. They are able to consider many points in a search space simultaneously and multiple searches in different area of fitness lans scape can be conducted at the same time, while conventional search methods use a single point. This ability, provide a high chance of global convergence and make EAs more powerful than traditional methods.

Genetic algorithm (GA) is recognized as the most well-known class of evolutionary algorithms. Genetic algorithms are stochastic algorithms inspired by Darwin's theory of evolution. Holland, in 1975, introduced a genetic algorithm to cope with combinatorial problems for the first time, and since then they have rapidly been implemented as one of the most powerful and efficient stochastic solution search procedures for solving several network design problems [34, 35]. Genetic algorithms are not dealing with decision variables and they do not need any domain knowledge of the problem. They use a structure to employ genetic information in order to find new search directions. They are based on coding and objective function for evaluating fitness. A population of feasible solutions is generated. Typically, these solutions are in the form of a string or chromosome. A selection strategy is applied to choose parents from the population. Genetic algorithms are implemented through genetic operators. The main genetic operators

First segment: random path-based direct encoding													Second segment: extended encoding for direct shipment and direct delivery														
Co	Di	P	R	Dc	S	...	Co	Di	P	R	Dc	S	...	Co	Di	P	R	Dc	S	P ₁	P ₂	...	P _i	Dc ₁	Dc ₂	...	Dc _x
3	2	2	5	6	2	...	3	1	4	3	1	2	...	4	1	3	5	2	1	1	0	...	2	1	1	...	0
Delivery path to customer 1						Delivery path to customer 2						Delivery path to customer M						Assigning the plants				Assigning the distribution centers					

Fig. 3 Representation of extended random path-based direct encoding method

observed in nature are reproduction, cross-over, and mutation [35, 36]. Offspring solutions are created with some of the characteristics of each parent based on these genetic operators. These operators (mutation and cross over) are named as the input variables of GA. It should also be noted that the crossover operator is usually applied as the main Genetic operator. It has the main role in the performance of a genetic algorithm, while a mutation operator is used as a background operator. Without cross over, all operator is used as a background operator. Without cross-over, all we have is local mutation. This means that, changes will happen slowly, and it will be very hard to get the population out of a local optimum. We can make a pretty huge jump from either of the parents by applying cross over. Mutation operators concentrate on the surroundings of a solution and make random small changes in chromosomes. It is not applied to make a big move in an unknown region but to move slowly [37].

Applying a population of solutions in GA allows for searching in multiple directions. But lack of enough intensification spurs researchers to another alternative with stronger search operators. According to [38, 39], memetic algorithms (MAs) have the benefits of GAs and also an additional local search engine to improve intensification. Recently, memetic algorithms (MAs) obtained increasing attention from the evolutionary computation community (ECC) and has been shown to be promising and effective with considerable success for solving difficult optimization problems in numerous applications domains [40–42]. Therefore, memetic algorithm has become a popular approach for various engineering optimization problem [42–45].

The utilized algorithm in this study is designed to establish the optimum link between the opened facilities and aims to minimize the cost while satisfied demand. Combining the aforementioned components, the flow of the proposed algorithm is as follows:

- Step 1:** (Data reading) Set parameters and read in the data of a given instance.
- Step 2:** (Initialization) Create an initial random population by the extended random path direct encoding method (defined in [16], Figs. 3, 4 and 5.). Repeat this process according to the population size denote by *N*. Each indi-

First segment							Delivery path	The delivery and recovery path				
Reverse flow			Forward flow									
Co	Di	P	R	Dc	P	S						
3	2	2	5	6	2	2	→	1	S ₂ - P ₂ - Dc ₆ - R ₅ - C ₁ - Co ₃ - Di ₂ /P ₂			
Delivery path to customer 1								2	S ₂ - P ₂ - R ₅ - C ₁ - Co ₃ - Di ₂ /P ₂			
								3	S ₂ - P ₂ - Dc ₆ - C ₁ - Co ₃ - Di ₂ /P ₂			
								4	S ₂ - P ₂ - C ₁ - Co ₃ - Di ₂ /P ₂			

Fig. 4 Delivery path for a sample of gene unit

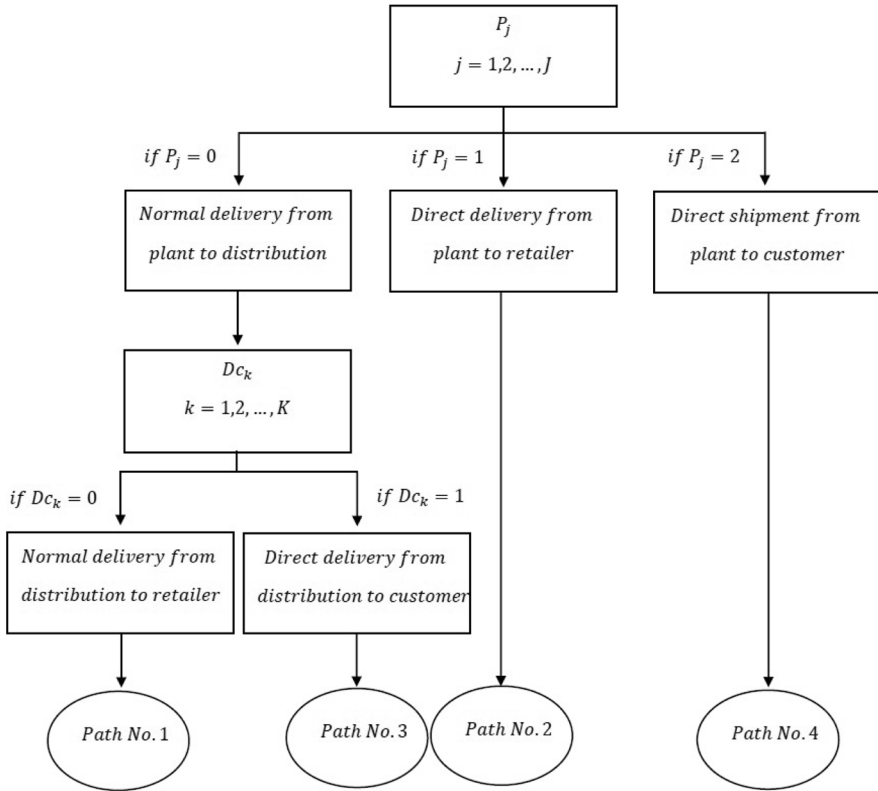


Fig. 5 Presentation of the second segment of the extended random path-based direct encoding

vidual in this population identify a supply chain configuration for the presented problem.

Step 3: (Evaluation) Calculate the fitness value of each individual in the population using the objective functions presented in 1.

Step 4: (Selection) Apply selection operator using roulette wheel selection.

Step 5: (Cross-over) For each selected parent, apply two-point cross-over operation to create two offsprings by respecting to the cross-over probability.

Step 6: (Merging, sorting) The new population obtained by cross over operator is merged to the initial population. The fitness values of all new individuals are calculated and the new merged population is sorted according to their fitness value.

Step 7: (Segregation) The best *N* individuals are reserved as the initial population for the next generation.

Step 8: (local search) The first individual which is the best as well is selected for local search operator (defined in [16]). If an improvement in fitness value is occurred, the new individual is exchanged by the current one, otherwise the previous one is kept as a member of population for the next generation.

Step 9: (Termination) The algorithm runs until one of the stopping conditions is satisfied. The first stopping criteria is total number of iterations which controlled by the parameter “maxit”. In the second criterion, a parameter as maximum number of iterations without improvement is defined. If the pre-specified stopping conditions are not satisfied, the algorithm needs to go back to Step 4.

Also, we would like to note that a complete explanation regarding above items has been presented in [16]. In this study, a memetic algorithm with a local search engine and a new chromosome representation is considered. Evaluation of the performance of our memetic algorithm is divided to two parts in order to solve the test instances. The first one is applied for small size problems using commercial package (LINGO optimization software), and the second is employing a classical genetic algorithm as the second meta-heuristic algorithm. The procedure of the classical genetic algorithm is displayed through a flowchart in Figure 6 to clearly show the overall steps involved in the algorithms. To have a comparison between these two, a same structure is considered for both algorithms. The only different is the local search engine is replaced by mutation operator in the proposed memetic algorithm. In this regard in following, we consider some explanation regarding the mutation operator applied for the classical genetic algorithm.

The mutation operator can create offspring on individuals and aim to vary the solution to avoid GA lead to Local optimum. Mutation operator is repeated according to number of mutation rate. For applying this operator, single parent is selected and a random change is happened on the selected parent by modifying one or more gene values. As this operator causes a very small change, a very low rate is used most of the time [46].

For the proposed genetic algorithm, random mutation operator is used which simply select a gene at random and replace it with a random number from a feasible range. This method is selected according to the characteristic of the chromosomes.

4 Test Problems and Computational Results

Within this section, 13 numerical examples with different sizes are carried out to show the performance of the proposed MA. Since the framework of this research is not exactly the same with the previous studies, there is no benchmark model available for the proposed model in the area of closed-loop supply chain. Therefore, the size of the presented numerical examples are selected randomly using uniform distribution as shown in Table 1. Performance of the proposed MA is proved in two levels. In the first level, we employed LINGO17 to provide optimal results for small size problems. In the second level, a comparison between the classical GA and the proposed MA is considered under the same condition. Also, the structure of the GA applied in this study is the same by the proposed MA; however, the local search mechanism is replaced by the classical mutation operator.

The first seven test problems are small sized and the number of decisions variables are 98, 128, 209, 234, 468, 1006, and 1780, respectively, and the remaining

Fig. 6 The flow diagram of the classical GA

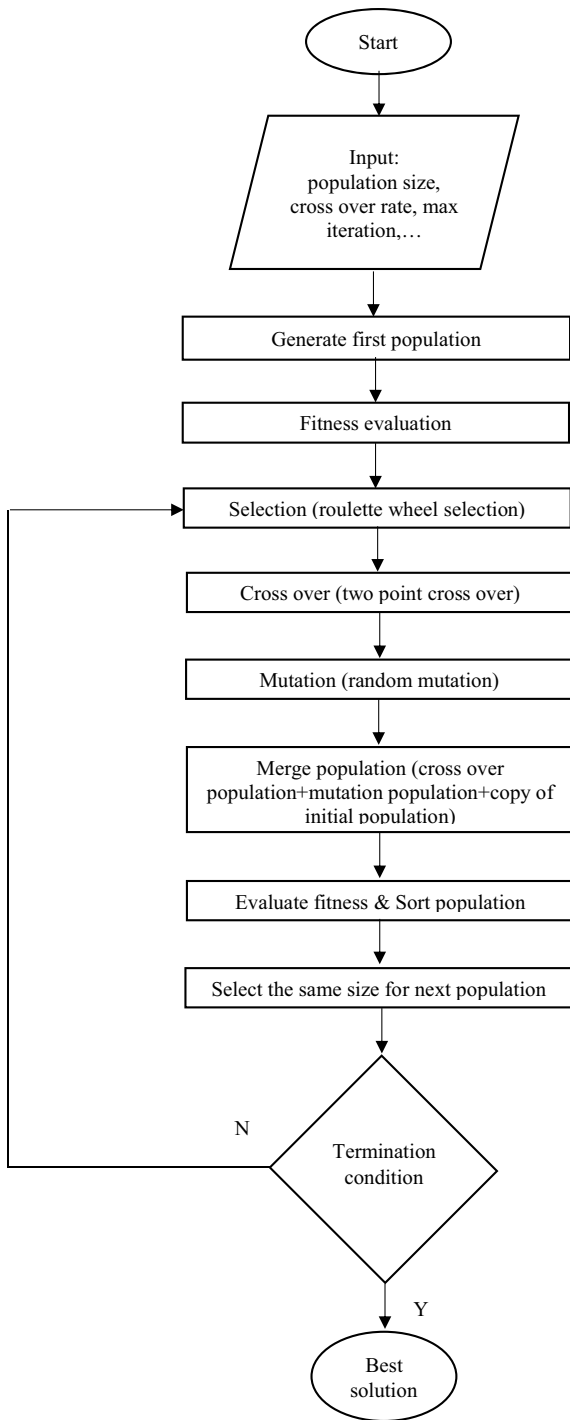


Table 1 Settings of test problems

Problem	S	P	Di	R	C	Co	Di
1	2	3	3	4	3	2	1
2	2	2	5	8	2	2	1
3	2	4	6	10	2	2	1
4	2	3	8	9	3	3	2
5	2	4	10	16	4	4	2
6	3	6	15	24	6	6	2
7	4	8	20	32	8	8	4
8	6	12	30	48	12	12	6
9	6	14	32	54	14	14	6
10	8	16	40	64	16	16	8
11	10	18	36	80	18	18	10
12	10	20	40	84	20	20	10
13	12	24	40	96	24	24	12

problems are large sized. Other parameters are generated randomly using uniform distributions according to the information presented in Table 2. To test the robustness of the method each test problem has been implemented 10 times on a personal computer with Intelr CoreTM i5 2.40 GHz with 12 GB RAM. The

Table 2 Parameters values used in the test problems

Parameters	Range
Capacity of facilities	
$b_j, j \in S$	Uniform (200,1100)
$b_j, j \in P$	Uniform (100,1000)
$b_j, j \in Dc$	Uniform (50,900)
$b_j, j \in R$	Uniform (50,850)
$b_j, j \in D$	Uniform (100,500)
$b_j, j \in Co$	Uniform (20,100)
$b_j, j \in Di$	Uniform (20,100)
Returned and disposal rate	
p_j^{return}	10 %
p_j^{disposal}	50 %
Transportation cost	
c_{ij}	Uniform (1,3)
Operation cost	
$c_j, j \in P$	Uniform (100,2500)
$c_j, j \in Dc$	Uniform (100,2100)
$c_j, j \in R$	Uniform (100,400)
$c_j, j \in Co$	Uniform (100,500)
$c_j, j \in Di$	Uniform (50,400)

Table 3 Results obtained by LINGO

Problem	Problem size	number of involved facilities	Solution	Ave time (s)
1	2 · 3 · 3 · 4 · 3 · 2 · 1	18	1920	0.05
2	2 · 2 · 5 · 8 · 2 · 2 · 1	22	2905	0.1
3	2 · 4 · 6 · 10 · 2 · 2 · 1	27	2345	0.12
4	2 · 3 · 8 · 9 · 3 · 3 · 2	30	2335	0.12
5	2 · 4 · 10 · 16 · 4 · 4 · 2	42	1160	0.14
6	3 · 6 · 15 · 24 · 6 · 6 · 2	62	4100	0.16
7	4 · 8 · 20 · 32 · 8 · 8 · 4	84	11365	0.17
8	6 · 12 · 30 · 48 · 12 · 12 · 6	126	-	
9	6 · 14 · 32 · 54 · 14 · 14 · 6	140	-	
10	8 · 16 · 40 · 64 · 16 · 16 · 8	168	-	
11	10 · 18 · 36 · 80 · 18 · 18 · 10	190	-	
12	10 · 20 · 40 · 84 · 20 · 20 · 10	204	-	
13	12 · 24 · 40 · 96 · 24 · 24 · 12	232	-	

proposed MA and GA were developed in the MATLAB 2016 on the above mentioned system.

The test problems are solved by the proposed MA under a constant population size of 100 while three different population sizes: 100, 200, and 300 are considered for the classical GA. Regarding stopping condition, we imposed a maximum iteration number of 200 as well as a maximum number of iteration without improvement 6, 8, 10, 12, 20, 25, and 30 for our small size problems, respectively. For the large-size problems, we increased the latter bound by 5. Also, to standardize all test problems we set the number of local search iterations to be

Table 4 Results for the proposed MA Algorithm with $n = 100$ and $crrate = 0.4$ over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	4.06
2	2905	2905	2905	5.11
3	2345	2405	2351	10.34
4	2335	2535	2355	32.7
5	1160	1340	1185	35.05
6	4100	4600	4222	97.6
7	11365	12095	11814	261.5
8	16588	18189	17300.5	2070
9	12358	15133	13330.6	2633.3
10	21585	26332	23457.8	3970
11	22056	23088	22400	10200
12	27388	29171	27982.3	14666.6
13	29039	31607	29982.3	26333.3

Table 5 Results for GA Algorithm with $n = 100$ and $crate = 0.4$ over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	1.75
2	2905	2905	2905	2
3	2345	2405	2369	3.1
4	2335	2535	2415	7.59
5	1160	1530	1225	10.2
6	4190	4720	4350	20.2
7	11805	12895	12292.3	50.1
8	18324	21974	19529.6	392
9	15430	16059	15706	583.3
10	25538	31389	28033.2	828
11	25488	27574	26660	1003.3
12	31746	34649	33681.33	1983.3
13	34103	43757	37667.2	2000

equal to the number of retailers for each test problem. Therefore the local search iteration number will be proper with the size of the test problems.

To evaluate the performance of the proposed MA and GA, LINGO17 is adapted to solve the optimization problems. Obtained results are presented in Table 3. Although LINGO provides optimal results for small size problems quickly, Table 3 indicates that LINGO is inappropriate for solving the large-size problems, and it is run out of memory.

According to the results presented in Table 4, for the test problem No. 1 and 2, as the equality of Min, Max and Average cost shows, the proposed MA is able to find the optimal solution for all 10 runs. For test problems No. 3 to 7, the value of Min cost indicates that the MA is capable to reach optimal solution. Therefore, we conclude that the proposed MA is able to provide good solutions for the small size problems, which allows us to trust the algorithm for large-size problems as well. The results obtained by the classical GA with three different sizes are presented in Table 5 to 7.

To indicate the trade-offs between the proposed MA and the classical GA, four comparison criteria are considered. First the relative gap between the mean of the objective function value's associated with the MA and the optimal solution obtained by LINGO optimization software (i.e., $OFV - Gap_{MA-LIN}$). Second, the same procedure for the classical GA (i.e., $OFV - Gap_{GA-LIN}$). Third, the relative between the mean of the objectives value associated with the MA and GA (i.e., $OFV - Gap_{GA-MA}$) and finally the relative gap between the mean of the CPU time's associated with the MA and GA (i.e., $CT - Gap_{MA-GA}$). The comparison is focused on two aspects: objectives function value and CPU time.

$$OFV - GAP_{MA-LIN} = \frac{MA_{answer} - LIN_{answer}}{LIN_{answer}} \tag{5}$$

Table 6 Results for GA Algorithm with $n = 200$ and $errate = 0.4$ over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	2.5
2	2905	2905	2905	3.2
3	2345	2405	2363	6.1
4	2335	2535	2395	13.36
5	1160	1360	1194	18.9
6	4100	4720	4279	42.3
7	11775	12425	12071	110.5
8	16956	18659	17742.5	744.5
9	13444	15321	14259	1126.6
10	22399	27525	25597.4	1507
11	25275	27503	26038	2000
12	30985	34602	32486	3183.3
13	33596	37277	35051.2	4010

$$OFV - GAP_{GA-LIN} = \frac{GA_{answer} - LIN_{answer}}{LIN_{answer}} \tag{6}$$

$$OFV - GAP_{GA-MA} = \frac{GA_{answer} - MA_{answer}}{MA_{answer}} \tag{7}$$

$$CT - GAP_{MA-GA} = \frac{CT_{MA}}{CT_{GA}} \tag{8}$$

Table 4 to 7 are demonstrated a comparison between the proposed MA and the classical GA. In order to show the performance of the proposed MA,

Table 7 Results for GA Algorithm with $n = 300$ and $errate = 0.4$ over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	4
2	2905	2905	2905	4.4
3	2345	2405	2357	9.5
4	2335	2535	2375	26.1
5	1160	1340	1192	27.4
6	4100	4720	4239	84.1
7	11475	12315	11933	203.3
8	16588	18625	17410	1333
9	13166	14293	13885	2016.6
10	21686	26396	24471.6	2590
11	24827	25544	25081.3	3700
12	30142	31871	31154	6456.2
13	33006	35807	34807	8410

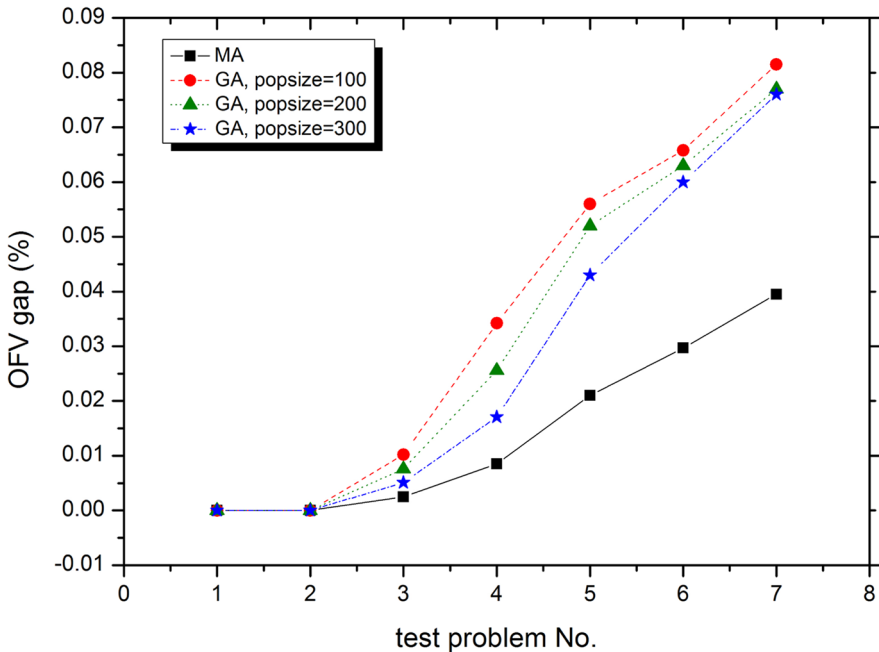


Fig. 7 Comparison of results from LINGO and the proposed MA and GA

objective function value gap is calculated by formula (1) based on the optimal solution obtained by LINGO optimization software and average cost computed by the MA. In a similar way the objective function gap is computed by formula (2) for the GA. By assessing the results presented in Table 8, we can observe the average $OFV - Gap_{GA-LIN}$ for the first 7 test problems which are small sized is 3.5 percent which is almost triple compared with the result reported by the MA (i.e., 1.4) (under the same population size of 100 for both algorithm). By increasing the population size to 200 and 300 this number decreased to 3.2 and 2.8, respectively.

From another point of view, as it is bold in Table 5, although the classical GA is able to find the optimal solution in each run for test problem No. 1 and 2 and reach the optimal solutions for test problem No. 3 and 4 during 10 runs, is not able to find the optimal solution for test problem number 6 and 7 which our proposed MA is able to find (Table 4). By increasing the population size to 200, the GA can find optimal solution for test problem number 6 but the problem is still remain for test problem number 7 even by increasing the population size to 300 (Table 6 and 7). But as shown, the Min cost for test problem No. 7 dropped from 11775 to 11475 by increasing population size from 200 to 300 and for average cost, dropped from 12071 to 11933 which show progress in result. Also for large-sized test problems, we can see improvement in Min, Max, and average cost by increasing population size.

A similar comparison for $OFV - Gap$ is provided in Figure 7 between the proposed Ma and the classical GA based on the LINGO optimization software listed

Table 8 Comparison of results from LINGO and the proposed MA and GA

Test problem	<i>OFV – GAP</i>			
	<i>MA – LINGO</i>		<i>GA – LINGO</i>	
	<i>popsize = 100</i>	<i>popsize = 100</i>	<i>popsize = 200</i>	<i>popsize = 300</i>
1	0	0	0	0
2	0	0	0	0
3	0.0025	0.0102	0.0076	0.0051
4	0.0085	0.0342	0.0256	0.0171
5	0.021	0.056	0.052	0.043
6	0.0297	0.0658	0.063	0.06
7	0.0395	0.0815	0.077	0.076
average (%)	1.4	3.5	3.2	2.8

in Table 8. Except test problem No.1 and 2 which are too small size problems the proposed MA present a good quality solution in comparison with the GA. When the MA and GA have the same size of 100, the *OFV – Gap* of the GA is always equal or higher than MA. By increasing population size to 200 and 300 an improvement in the GA is observed, however, they are not really competitable with the results obtained by the proposed MA and just slightly improve the results of the GA. We need to mention that in this study we kept the population size of the MA the same for all runs. Undoubtedly, higher population size for the MA let the algorithm to provide search mechanism stronger and find better results.

The results in Table 4 and 5 obtained based on the equal population size of 100 for the MA and GA. The average *OFV – Gap_{GA-MA}* presented in Table 9 obtained through the mentioned condition calculated by formula (3) is 9.8%, which shows

Table 9 Comparison of objective function value between the MA and GA

Test problem	<i>OFV – GAP_(GA-MA)</i>		
	<i>popsize = 100</i>	<i>popsize = 200</i>	<i>popsize = 300</i>
1	0	0	0
2	0	0	0
3	0.007	0.005	0.002
4	0.025	0.016	0.008
5	0.033	0.03	0.02
6	0.035	0.032	0.03
7	0.04	0.036	0.035
8	0.12	0.051	0.038
9	0.17	0.069	0.041
10	0.19	0.091	0.043
11	0.2	0.16	0.11
12	0.2	0.16	0.11
13	0.256	0.169	0.160
average (%)	9.8	6.3	4.6

Table 10 Comparison of CPU time between the MA and GA

Test problem	$CT - GAP_{(MA-GA)}$		
	$popsize = 100$	$popsize = 200$	$popsize = 300$
1	2.32	1.62	1.015
2	2.555	1.59	1.16
3	3.33	1.69	1.08
4	4.3	2.44	1.25
5	3.43	1.854	1.27
6	4.83	2.30	1.16
7	5.219	2.36	1.28
8	5.28	2.78	1.55
9	4.51	2.33	1.30
10	4.79	2.63	1.532
11	10.1	5.1	2.756
12	7.3	4.6	2.27
13	13.16	6.5	3.13
average	5.47	2.9	1.5

a noticeable difference in the objective function value between these two algorithms. To improve the performance of the GA, population size is increased to 200. According to the results of the Table 9 and 10, this change causes the average $OFV - Gap_{GA-MA}$ to be decreased to 6.3% while the average $CT - Gap_{MA-GA}$ calculated by formula (4) was almost doubled (Table 10) but still a significant gap can be seen between the $CT - Gap$ of the MA and GA. To have more improvement, the population size of the GA is increased up to 300. The results show the average $OFV - Gap_{GA-MA}$ is reduced to 4.6 but the average $CT - Gap_{MA-GA}$ was again doubled (Table 10). Also from the Table 10 we realized our proposed MA needs longer computation time compare by the GA when they both run under the same population size of 100. But as we are dealing with strategic problem in this study, the quality of solution has priority than run time. Therefore, although it is obvious that GA is much better than the proposed MA in run time, it fails to find good solutions for large-sized problems. Since the size of real case studies are large, the proposed MA is more sufficient.

A same comparison for $OFV - Gap$ is shown in Figure 8 between the proposed Ma and the classical GA with three different population sizes, provided in Table 9. Although increasing in population size made a good improvement in the performance of the GA, do not show the GA as a compatible algorithm to the proposed MA.

From another aspect, Figure 9 demonstrates an example of the proposed MA and GA convergence during 100 iteration related to problem 7. This figure reveals that the convergence of the MA is very sharp in comparison with the GA. It means the MA can easily reach to the good area in search space and do a good search and improve the results and need less iterations to find better solutions compare by the GA. The proposed MA found the optimal solution after 70 iteration while the GA

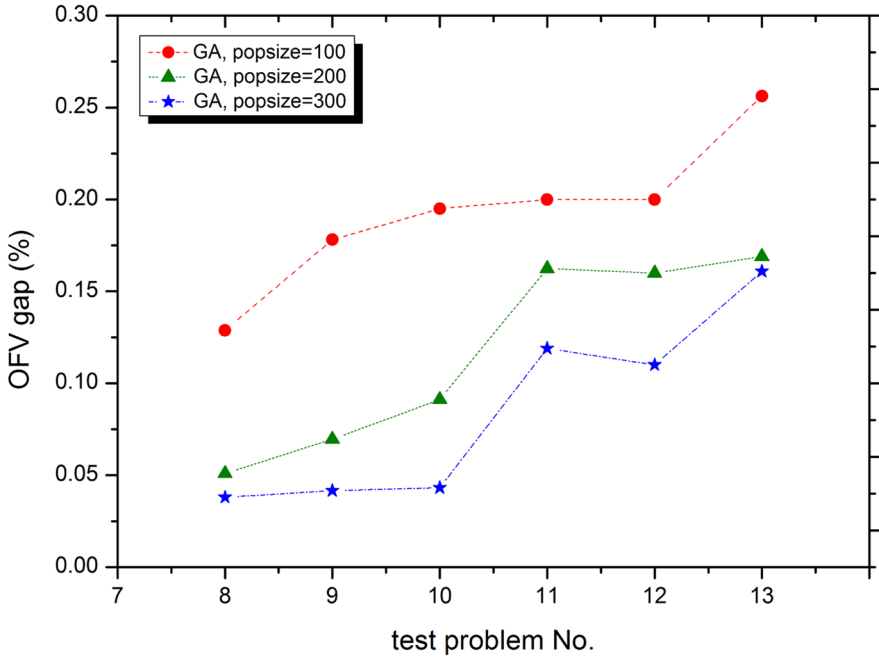


Fig. 8 Comparison between the proposed MA and classical GA

could not find the optimal solution in 100 iteration even by increasing population size to 300.

General speaking, the obtained results prove that the GA cannot find an appropriate near optimal solution in large-size problems for the proposed flexible integrated forward/reverse logistics network without enriching the algorithm with a powerful local search engine. For the presented network with a complex nature, lack of intensification of the classical GA prevent the algorithm to find a good solution. We showed that the proposed MA algorithm is able to efficiently find a good solution for small and large-size problems.

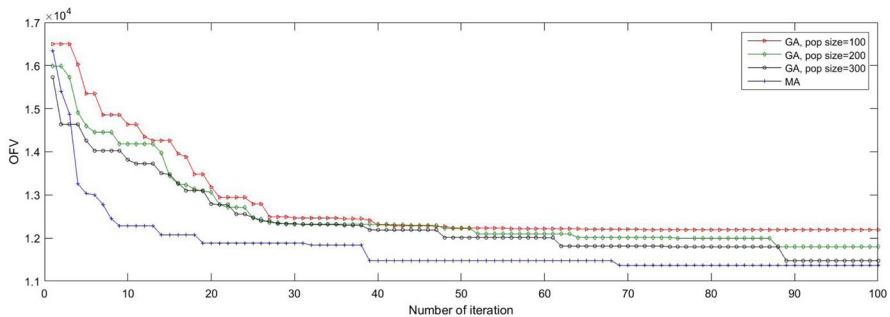


Fig. 9 Comparison of the convergence between the proposed MA and classical

5 Conclusions and Future Studies

We focused on a comprehensive mixed integer linear programming formulation for a seven stages closed-loop network design problem. We applied the extended random path direct encoding method-based memetic algorithm, which was developed for a full delivery graph and a combined forward/reverse logistics design to decrease delivery time and avoid suboptimal solutions, respectively. The aim of this work is to minimize total cost, which we addressed as allocation problem to find the optimal number and capacity for any facility as well as the optimal transportation flow between facilities. Since the basic problem is NP-hard, the combination with flexibility in delivery path makes the search space of the problem much larger and more complex and NP-hard as well. Because existing methods are unable to solve this problem, we proposed a MA approach to compute a near optimal solution for large-size problems. Within this study, we treat several numerical examples to verified correctness of the proposed method as well as confirm the effectiveness of that using a commercial package and a classical GA. On the other hand we applied the commercial solver, to show the performance of the classical GA. Considering the scale of the test problems, the results display that the proposed memetic algorithm can effectively detect solutions that are close to optimal. Also the $OFV - Gap$ between the GA and MA is significantly large which shows the efficiency and capability of the proposed MA.

Also, it was observed that the evaluation speed in the MA is faster while in GA is slow, even by increasing population size. Increasing population size does not show a significant improvement in the results obtained by the GA, while the computational time is almost double from 100 to 200 and from 200 to 300. In contrast, the proposed MA needs more computational time than the GA and can be considered as a time consuming algorithm.

At the end, the results confirm that the classical GA cannot obtain a good solution without applying a powerful local search engine, because of a complex nature of the proposed flexible integrated forward/reverse logistics network while our MA approach produce high-quality solution. Therefore we believe the presented method will be an efficient method to solve this kind of multi-stage logistics network design problems.

Apart from costs aspect considered here, other aims such as responsiveness and robustness can be considered in designing integrated forward/reverse logistics network that needs updating the algorithm to be capable to solve multi-objective models. Moreover, to be close to the real world application, multi-product multi-capacity and multi-period networks with uncertainty as well as considering inventory can be employed.

Acknowledgements The authors would like to appreciate the support International Graduate School (IGS) of Bremen University to support, help and advice as well as the Deutscher Akademischer Austausch Dienst (DAAD) for financial support of this research under the GSSP programme of the IGS.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Statement The data that support the findings of this study are available from the corresponding author on request.

Declarations

Conflict of Interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Neto JQF, Walther G, Bloemhof J, Nunen JAEEV, Spengler T (2009) From closed-loop to sustainable supply chains: the weee case. *Int J Prod Res* 48(15):4463–4481
2. Aras N, Aksen D, Tanugur A (2008) Locating collection centers for incentive dependent returns under a pick-up policy with capacitated vehicles. *Eur J Oper Res* 191:1223–1240
3. Alumur SA, Nickel S, da Gamad FS, Verter V (2012) Multi-period reverse logistics network design. *Eur J Oper Res* 220(1):67–78
4. Maslennikova I, Foley D (2000) Xerox's approach to sustainability. *Interfaces* 30:226–233
5. Lee D, Dong M (2007) A heuristic approach to logistics network design for end-of lease computer product recovery. *Transp Res Part E* 44:455–474
6. Pishvae MS, Farahani RZ, Dullaert W (2010) A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Comput Oper Res* 37:1100–1112
7. Kilibi W, Martel A, Guitouni A (2010) The design of robust value-creating supply chain networks: a critical review. *Eur J Oper Res* 203:283–293
8. Altiparmak F, Gen M, Lin L, Paksoy T (2006) A genetic algorithm approach for multi-objective potimization of supply chain networks. *Comput Ind Eng* 51:197–216
9. Du F, Evans G (2008) A bi-objective reverse logistics network analysis for post-sale service. *Comput Oper Res* 35:2617–2634
10. Yeh W (2005) A hybrid heuristic algorithm for the multistage supply chain network problem. *Int J Adv Manuf Technol* 26:675–685
11. Jayaraman V, Patterson R, Rolland E (2003) The design of reverse distribution networks: mmodel and solution pprocedure. *Eur J Oper Res* 150:128–149
12. Pishvae MS, Kianfar K, Karimi B (2010) Reverse logistics network design simulated annealing. *Int J Adv Manuf Technol* 47(1):269–281
13. Yadegari E, Zandieh M, Najmi H (2015) A hybrid spanning tree-based genetic/simulated annealing algorithm for a closed-loop logistics network design problem. *Int J Appl Decis Sci* 8
14. Yadegari E, Ekhtiari M, Zandieh M, Alem-Tabriz A (2014) An artificial immune algorithm for a closed-loop supply chain network design problem with different delivery paths. *Int J Strateg Decis Sci* 5(3)
15. Melo M, Nickel S, da Gama FS (2009) Facility location and supply chain management: a review. *Eur J Oper Res* 196:401–412
16. Behmanesh E, Pannek J (2016) A memetic algorithm with extended random path encoding for a closed-loop supply chain model with flexible delivery. *J Logist Res* 9(22):1–12
17. Yun Y, Moon C, Kim D (2009) Hybrid genetic algorithm with adaptive local search scheme for solving multistage-based supply chain problems. *Comput Ind Eng* 56:821–838

18. Moghadam RT, Safaei N, Sassani F (2009) A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Comput Oper Res* 36(2):402–414
19. Lin C, Chen K, Chuang W (2012) Motion planning using a memetic evolution algorithm for swarm robots. *Int J Adv Rob Syst* 9:1–9
20. Ko HJ, Evans GW (2007) A genetic-based heuristic for the dynamic integrated forward/reverse logistics network for 3pls. *Comput Oper Res* 34:346–366
21. Wang Y, Ma X, Li Z, Liu Y, Xu M, Wang Y (2017) Profit distribution in collaborative multiple centers vehicle routing problem. *J Clean Prod* 144:203–219
22. Wang Y, Peng S, Zhou X, Mahmoudi M, Zhen L (2020) Green logistics location routing problem with eco packages. *Transp Res Part E: Logist Transp Rev* 143:102–118
23. Wang Y, Yuan Y, Guan X, Xu M, Wang L, Wang H, Liu Y (2020) Collaborative two-echelon multi-center vehicle routing optimization based on state-space-time network representation. *J Clean Prod* 258:120–590
24. Fu X, Pace P, Aloï G, Yang L, Fortino G (2020) Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm. *Comput Netw* 177:107–327
25. Fu X, Yao H, Postolache O, Yang Y (2019) Message forwarding for wsn-assisted opportunistic network in disaster scenarios. *J Netw Comput Appl* 137:11–24
26. Fu X, Yao H, Yang Y (2019) Exploring the invulnerability of wireless sensor networks against cascading failures. *Inf Sci* 491:107–327
27. Grag P (2009) A comparison between memetic algorithm and genetic algorithm for the cryptanalysis of simplified data encryption standard algorithm. *Int J Netw Secur Appl* 1:34–42
28. Yadegari E, Alem-Tabriz A, Zandieh M (2019) A memetic algorithm with a novel neighborhood search and modified solution representation for closed-loop supply chain network design. *Comput Ind Eng* 128:418–436
29. Syarif A, Yun YS, Gen M (2002) study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach. *Comput Ind Eng* 43(1):299–314
30. Afkar A, Kaleibar MM, Payani A (2012) Geometry optimization of double wishbone suspension system via genetic algorithm for handling improvement. *J Vibroengineering* 14:827–883
31. Richard MJ, Bouazara M, Khadir L, Ca GQ (2011) Structural optimization algorithm for vehicle suspensions. *Trans Can Soc Mech Eng* 35:1–17
32. Yaman F, Yilmaz AE (2010) Impacts of genetic algorithm parameters on the solution performance for the uniform circular antenna array pattern synthesis problem. *J Appl Res Technol* 8(3):378–439
33. Vargas-Martinez A, Garza-Castanon LE (2011) Combining artificial intelligence and advanced techniques in fault-tolerant control. *J Appl Res Technol* 9(2):202–222
34. Gen M, Cheng R (2000) Genetic algorithms and engineering optimization. Wiley, New York
35. Gen M, Cheng R, Oren SS (2001) Network design techniques using adapted genetic algorithms. *Adv Eng Softw* 32:731–744
36. Hendriks M, Voeten B, Kroep L (1999) Human resource allocation in a multi-project r and d environment. *International J Prod Mgmt* 17(2):181–188
37. Kim K, Yun Y, Yoon J, Gen M, Yamazaki G (2005) Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Comput Ind* 56:143–160
38. Behmanesh E, Pannek J (2016) Modeling and random path-based direct encoding for a closed loop supply chain model with flexible delivery paths, in the seventh IFAC conference on Management and Control of Production and Logistics, p. submitted
39. Lin YC (2013) Mixed-integer constrained optimoptimization on memetic algorithm. *J Appl Res Technol* 11:242–250
40. Hart WE, Krasnogor N, Smit JE (2005) Recent Advances in Memetic Algorithms. Springer-Verlag
41. Ishibuchi H, Yoshida T, Mura T (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans Evol Comput* 7(2):204–223
42. Tang M, Yao X (2007) A memetic algorithm for vlsi floorplanning. *IEEE Trans Syst* 37(1):62–69
43. Mei Y, Tang K, Yao X (2011) Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem. *IEEE Trans Evol Comput* 15(2):151–165
44. Ahn Y, Park J, Lee C, Kim J (2010) Novel memetic algorithm implemented with ga (genetic algorithm) and mads (mesh adaptive direct search) for optimal design of electromagnetic system. *IEEE Transp* 46(6):1982–1985
45. Li B, Zhou Z, Zou W, Li D (2012) Quantum memetic evolutionary algorithm-based low-complexity signal detection for underwater acoustic sensor networks. *IEEE Trans Evol Comput* 42(5):626–640

46. Gen M, Kumar A, Kim JR (2005) Recent network design techniques using evolutionary algorithms". *Int J Prod* 98:251–261

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.