DOI 10.1007/s10845-015-1055-3

A differential evolution algorithm for the capacitated VRP with flexibility of mixing pickup and delivery services and the maximum duration of a route in poultry industry

Darat Dechampai · Ladda Tanwanichkul · Kanchana Sethanan · Rapeepan Pitakaso

Received: 18 April 2014 / Accepted: 8 February 2015 / Published online: 20 February 2015

Abstract In this paper, we propose two heuristics to solve the General Q-Delivery Vehicle Routing Problem with consideration of flexibility of mixing pickup, delivery services and a maximum duration of a route constraint which is the extending version of the well-known VRP with pickup and delivery problem. Firstly, the heuristic called DE_G-Q-DVRP-FD is presented to determine the routing of transferring pullets from pullet houses to hen houses. Since the problem considered is very complicated, the DE_G-Q-DVRP-FD is extended to the two-phase heuristic called MESOMDE_G-Q-DVRP-FD. The difference between two heuristics is that in the MESOMDE_G-Q-DVRP-FD algorithm, the customer vertices (pullet houses) will be clustered before determining routes. The clustering of customer vertices method called the Multifactor Based Evolving Self-Organizing Map is proposed in the first phase in order to completely utilize the vehicle. Finally, in the second phase, the DE_G-Q-DVRP-FD is used to execute the routing. To demonstrate the algorithm efficiency, flock allocation from pullet houses to hen houses in the egg industry is used as the case study. The results obtained from this study show that the MESOMDE_G-Q-DVRP-FD algorithm provides lower total cost values than that of the firm's current practice by 7.59–31.28 and 0.84–13.15 % better than the DE_G-Q-DVRP-FD algorithm. Additionally, the MESOMDE_G-Q-DVRP-FD is adjusted to solve the benchmark problem found in the literature. The experimental results show that the MESOMDE_G-Q-DVRP-FD algorithm yields better total cost values by 5.72–61.60 % (with an average of 31.46 %).

Keywords Poultry houses · Hen egg production · Differential evolution · Self-Organizing Map · Total cost

Introduction

This paper considers a special vehicle routing problem with pickup and delivery services (PDP). This problem is found in various practical operations of real-world industries. This research focuses on the PDP in the egg industry as a case study application. In egg production, eggs are produced by batches of hens according to production schedules in order to meet customer demands. A firm generally starts by purchasing chicks from a hatchery. The chicks are then transferred to be raised in pullet houses, where they stay until the age of 17 weeks. Normally, the capacities of the pullet house are heterogeneous depending on investment level of contracted farms. These pullets are then moved to hen houses and fed to a prescribed body weight to support egg production, and remain in the hen house until they reach 75 weeks of age. Then the hens are slaughtered since they have diminishing egg-laying capacity. A flow chart of hen egg production is depicted in Fig. 1. Hence, egg production is considered a dynamic production leading to

D. Dechampai · K. Sethanan
Research Unit on System Modeling for Industry, Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand
e-mail: darat_dech@hotmail.com

K. Sethanan
e-mail: ksethanan@gmail.com

L. Tanwanichkul (✉)
Department of Civil Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand
e-mail: ladda.tan@hotmail.com

R. Pitakaso
Metaheuristics for Logistic Optimization Laboratory, Department of Industrial Engineering, Ubon Ratchathani University, Ubon Ratchathani, Thailand
e-mail: enrapepi@ubu.ac.th

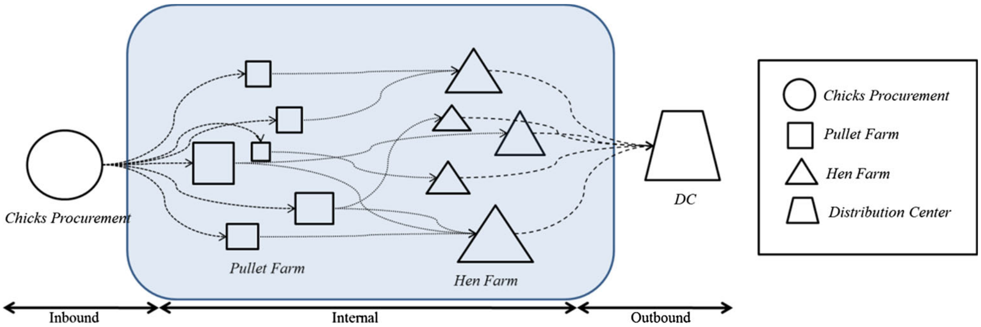


Fig. 1 The supply chain of egg production in the case study company (Dechampai et al. 2013)

complex production, especially the decision on how to allocate flocks to facilities.

Generally, the firm has difficulty to obtain efficient flock allocation, especially allocation of pullets to hen houses. In the flock transfer process, the vehicles must first pick up flocks at a pullet house and then deliver the flocks to a hen house. Due to an imbalance between the flocks to move out from a pullet house, the capacity of the fleet, and the available capacity of the hen house to move the flocks in that time period, a vehicle may pickup pullets from more than one pullet house until it reaches its capacity. It may then deliver pullets to more than one hen house while the maximum duration of a route constraint exists. Hence, allocation of flocks to facilities in the current period may affect the allocation efficiency of flocks in the next period. An inefficient flock allocation may lead to high transportation costs and vehicle utilization costs in terms of a higher number of vehicles used. To reduce such costs, in real practice, the vehicle may be allowed to have flexible mixed pickup and delivery services. Pickups can be accepted while not necessarily finishing all deliveries, and deliveries can be performed at any level of the commodity in the vehicle unless the delivery request is not satisfied (see Fig. 2). Hence, the flock allocation is related to the General Q-Delivery Traveling Salesman Problem (G-Q-DTSP) since its characteristics contain the following restrictions: (1) a single-commodity (i.e., pullets) ($p = 1$), (2) each pullet house is only for pickup operation while the hen house is only for delivering operation (P/D), (3) the depot does not supply (pick up) nor demand (deliver) any commodity units (pullets), and (4) each pullet house may be visited more than once in case the number of pullets to transfer is greater than the capacity of the vehicle. Therefore, the vehicle routes may not be restricted to Hamiltonian tours. The design of the vehicle routes for this problem is related to the split version of the pickup and delivery traveling salesman problem with a single commodity (1-PDTSP) allowing facilities (pullet houses and hen houses) to provide or demand any number of a commodity (pullets).

The G-Q-DTSP is a difficult and challenging problem in the field of vehicle routing problems (VRP). However, the problems faced by the egg production industry are more complex than the G-Q-DTSP in three aspects. (1) In order to meet the objective function by minimizing the total transportation costs consisting of two cost components: fuel cost and cost of vehicle utilization (i.e., vehicle rent cost), the vehicle is allowed to mix pickup services and delivery services in the vehicle routing while the maximum duration of a route constraint exists. (2) Due to imbalance between the flocks to move out from a pullet house, the capacity of the fleet, and the capacity of the hen house to move the flocks in, some vertices (i.e., pullet houses and hen houses) may have multiple visits. However, at each vertex, it is either a pick up or a delivery operation (P/D). This means the pullet house is a pickup service, while the hen house is a delivery operation. (3) Multiple vehicles are used to transfer pullets to hen houses ($k > 1$) based at the depot. Thus, in this paper, the transportation model is characterized as a "many-to-many structure: M–M" [i.e., several origins (pullet houses) and destinations (hen houses)] for only one commodity [M−M|P/D|k > 1]. It is rather a special G-Q-DTSP problem. In this study, the problem can be considered as the General Q-Delivery Vehicle Routing Problem (G-Q-DVRP) variant with the flexibility of mixing pickup and delivery services with constraints on maximum duration of a route. Since the Q-DTSP including the TSP is NP-hard (Berbeglia et al. 2007), in this paper, the G-Q-DVRP with consideration of flexibility of mixing pickup and delivery services with constraints on maximum

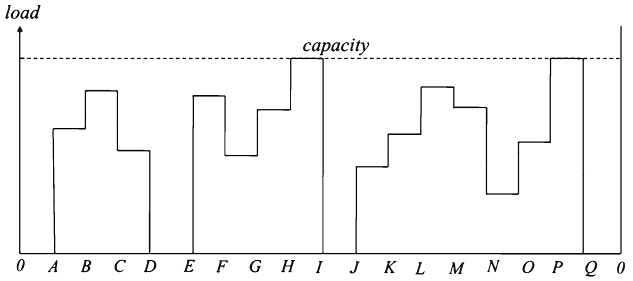


Fig. 2 Characteristics of flexibility of mixing pickup and delivery services of the problem

duration of a route (G-Q-DVRP-FD) is more complex than the Q-DTSP problem and remains NP-hard. Therefore, this observation makes a strong case for the application of heuristics and meta-heuristics to solve the problem.

This research focuses on determining routes for transferring pullets from pullet houses to hen houses with the objective to minimize the total transportation costs with the consideration of capacity restriction, maximum duration of a route, and multiple depots. The total cost consists of two cost components: fuel cost and cost of vehicle utilization. The fuel cost can be minimized by reducing the traveling distance while the cost of vehicle utilization can be reduced by minimizing the number of vehicles by fully utilizing the vehicles' capacity. To solve the problem, two heuristic are proposed. The first heuristic called Differential Evolution for G-Q-DVRP with consideration of flexibility of mixing pickup and delivery services with constraints on maximum duration of a route (DE_G-Q-DVRP-FD) uses the differential evolution technique (DE). Since the problem is very complicated, a two-phase algorithm called Multifactor Based Evolving Self-Organizing Maps with Differential Evolution for G-Q-DVRP with consideration of flexibility of mixing pickup and delivery services and the maximum duration of a route constraint (MESOMDE_G-Q-DVRP-FD) is developed. In the first phase, clustering of pullet houses was determined before allocating pullets to the hen houses efficiently. According to Dechampai et al. (2013), the Multifactor Based Evolving Self-Organizing Map (MESOM) was developed to cluster pickup customer vertices (i.e., pullet houses) in order to reduce traveling distance. The MESOM was introduced by modifying from the traditional SOM in that the MESOM considers not only the distance as the main factor but also the hen house capacity. Most importantly, the MESOM algorithm was developed for clustering pullet houses by the pullet capacity of each house that can be split and its coordinates that will be adjusted to two locations with splitting-coefficient of centroid movement in order to completely utilize the vehicle capacity. Due to its effectiveness, the MESOM was therefore adopted to cluster pickup customer vertices (i.e., pullet houses) in the first phase. The differential evolution technique (DE) developed in the DE_G-Q-DVRP-FD algorithm was used to determine routes for transferring pullets from the pullet houses to the hen houses in the second phase. To illustrate the effectiveness of the algorithm, numerical experimental results were compared with those of the current practice of the egg industry selected as a case study. Additionally, the MESOMDE_G-Q-DVRP-FD was evaluated by using the benchmark approach called the Growing Neural Gas (GNG) (Boonmee et al. 2013). In the next section, related literature is reviewed. The methodology used to solve the problem is presented in "Methodology" section. "Case Study Application" section presents the case-study application and illustrates numerical examples and performance of the algorithms. Finally a summary of the main findings is given in "Numerical experiments" section.

Literature review

Pickup and delivery problems (PDPs) constitute an important class of vehicle routing problems in which objects (commodities) or people have to be collected from the origins and distributed to the destinations. The PDP problems arise in various contexts such as logistics, ambulatory services, and robotics (Landrieu et al. 2001; Berbeglia et al. 2007; Ai and Kachitvichyanukul 2009; Kim and Son 2012). In the PDPs context, each vertex (i.e. customer), including the depot, can either require or supply a non-negative number of each commodity. The available vehicles to pick up and deliver commodities may have heterogeneous capacity. The load of each vehicle must not exceed its capacity. However, additional assumptions and constraints on vehicle routes may be presented depending on a specific problem, such as time windows, or multiple visits at some vertices. As a result, the vehicle routing problem arises in various contexts (see, e.g., Belmecheri et al. 2013; Vahdani et al. 2012; Wang and Lu 2010).

To understand the PDP models clearly, more recently, Berbeglia et al. (2007) have presented a general framework to model a collection of pickup and delivery problems as well as a three-field classification scheme [Structure:Visits:Vehicles]. The first field, named structure, specifies the number of origins and destinations of the commodities and is classified in three types: many-to-many problems (M–M) (see, e.g., Chen et al. 2014), one-to-many-to-one problems (1–M–1) (see, e.g., Erbao et al. 2008; Martinovic et al. 2008; Zachariadis et al. 2009), and one-to-one problems (1–1) (see, e.g., Şahin et al. 2013). In the M–M problems, any vertex can serve as a source or as a destination for any commodity. In one-to-many-to-one problems (1–M–1), commodities are initially available at the depot and are distributed to the customers. Additionally, commodities available at the customers are transferred to the depot. Finally, in one-to-one problems (1–1), each commodity has a given origin and a given destination. This type of problems arises, for example, in courier operations and door-to-door transportation services (see, e.g., Cordeau and Laporte 2003). The second field provides information on the way pickup and delivery operations are performed at customer vertices. Three notations are classified. Firstly, the notation PD indicates that each customer vertex is visited exactly once for a combined pickup and delivery operation, while it is noted as $P-D$ if these two operations may be performed together or separately. Lastly, the P/D notation indicates the case that either a pickup or a delivery operation is performed at each customer vertex.

The last field indicates the number of vehicles used in the problem.

The problem considered in this paper is related to the pickup and delivery problems with many-to-many structure (M–M). In the M–M structure, three problems are classified: the Swapping Problem (SP), the One-Commodity Pickup and Delivery Traveling Salesman Problem (1-PDTSP), and the Q-Delivery Problem respectively (Q-DTSP) (Berbeglia et al. 2007). The SP is a many-to-many with n-commodity PDP introduced by Anily and Hassin (1992a). The 1-PDTSP is a many-to-many, single-commodity problem firstly introduced by Hernandez-Perez and Salazar-Gonzalez (2004). A single vehicle based at the depot is considered to serve various customers classified into pickup and delivery customers. Additionally, the route of the vehicle is Hamiltonian. The 1-PDTSP is closely related to a special case of the SP with the additional constraint that the vehicle route is Hamiltonian (Berbeglia et al. 2007). For the Q-DTSP, only a single commodity is considered and is firstly introduced by Anily and Bramel (1999). Hence, the Q-DTSP is a special case of the 1-PDTSP that allows the customer vertices to provide or demand any number of commodities. A more general problem named the General Q-Delivery Traveling Salesman Problem (G-Q-DTSP), allows customer vertices to provide or demand any number of objects. The G-Q-DTSP is similar to the 1-PDTSP. The difference is that the vehicle route is not necessarily Hamiltonian. Therefore, the G-Q-DTSP is sometimes seen as a split version of the 1-PDTSP (Berbeglia et al. 2007). Moreover, the split delivery problem can be extended in the vehicle routing problem that a delivery to a demand vertex can be split between any number of vehicles (see, e.g., Dror and Trudeau 1989; Dror et al. 1994; Archetti et al. 2006; Dong et al. 2011; Şahin et al. 2013; Chen et al. 2014). Hence, in this study, the problem can be considered as the General Q-Delivery Vehicle Routing Problem (G-Q-DVRP) variant with the flexibility of mixing pickup and delivery services with constraints on maximum duration of a route. The problem has never been studied before (see Table 1).

To solve the problem, various methods have been applied. Over the last decade, DE as a novel evolutionary technique has been developed. Storn and Price (1997) first introduced the DE algorithm that was effectively applied to continuous optimization. DE is a population based search technique, which uses a simple operator with the classical operators of crossover, mutation and selection to create new candidate solutions. Its simple arithmetic operators, straightforward nature, quick convergence and features make it very attractive for numerical optimization. DE is one of the best evolutionary algorithms and is used extensively in various fields such as scheduling, machine layout, and manufacturing problems. For examples, in 2006, Nearchou used DE to design a machine layout problem, the unidirectional loop-layout design problem (LLDP) and to solve manufacturing problems with mixed integer discrete variables. In 2012 Vincent et al. proposed DE to schedule flexible assembly lines (FAL). Recently, Karen et al. (2013) used DE to describe how to use intelligent die design based on shape and topology optimization. In the same year, Chakaravarthy et al. (2013) presented DE and Particle Swarm Optimization algorithms (PSO) for scheduling m-machine flow shops with lot streamingand Ma et al. (2013) proposed a hierarchical hybrid particle swarm optimization (PSO) and differential evolution (DE) based algorithm (HHPSODE) to deal with bi-level programming problems (BLPP).

Although the DE has been effectively used in a variety of fields, it has been very limited when applying to solve VRP, especially the Vehicle Routing Problem with Pick-up and Delivery (VRPPD), because the encoding scheme of DE cannot be directly adopted for VRP. Erbao et al. (2008) presented a hybrid optimization algorithm (HOA), which is based on a combination of differential evolution (DE) and the genetic algorithm (GA). The advantages of both DE and GA are optimizing large scale problems effectively and the optimization process. In 2009, Erbao and Mingyong proposed stochastic simulation and the differential evolution algorithm to solve vehicle routing problems with fuzzy demand (VRPFD). The proposed differential evolution algorithm can also solve other deterministic counterparts in that the VRPFD considers more complicated side constraints. Later, Lai and Cao (2010) proposed an improved differential evolution algorithm (IDE) for solving vehicle routing problem with simultaneous pickups and deliveries and time windows (VRP-SPDTW). A novel decimal coding was adopted to construct an initial population and an integer order criterion based on natural number coding method. In the same year, Hou et al. (2010) presented a novel discrete differential evolution algorithm (DDE) for solving stochastic vehicle routing problems with simultaneous pickups and deliveries (VRPSPD) by developing a novel mutation operator which can be used in the discrete domain directly. That result obtains better results than the traditional DE algorithm and the existing GA algorithm.

However, customer vertices can be clustered before determining routes by several techniques such as K-mean (MacQueen 1967), GNG (Fritzke 1995) and hGNG (Boonmee et al. 2015). One of the most popular data clustering techniques is the self-organizing map (SOM) that is proposed by Kohonen (1995). The SOM is widely used in several applications such as resource allocation (Arnonkijpanich et al. 2004), data representation and compression (Arnonkijpanich et al. 2011a). The SOM conceptually works by dividing a large set of vectors into groups, and each group is then represented by its prototype (Dechampai et al. 2013). Furthermore, Deng and Kasabov (2000) proposed an algorithm of an evolving self-organizing map (ESOM), which is different from that of SOM. The network structure of ESOM evolves in an online adaptive mode. No topological constraint is given for the

Table 1 A comparison of the different papers on the VRPPD

Author (year)	Structure	Number of commodities	Visit type	Flexibility of mixing pickup and delivery	Pickup-first delivery-second	Mixed pickups and deliveries	Time constraint	Maximum duration of a route	Split delivery
Our problem	M–M	1	P/D	√				√	√
Chen et al. (2014)	M–M	>1	PD			√			√
Şahin et al. (2013)	1–1	1	P/D		√				√
Zachariadis et al. 2009	1–M–1	1	PD			√			
Martinovic et al. (2008)	1–M–1	1	P/D			√			
Erbao et al. (2008)	1–M–1	1	PD			√	√		
Nagy and Salhi (2005)	1–M–1	1	PD			√			

feature map a priori and the ESOM network starts without nodes. During learning, the network updates itself, creating new nodes when necessary. These algorithms construct the topology by using only the distance criteria. Additionally, the SOM can also be applied for enhancing agriculture supply chain. Arnonkijpanich et al. (2004) developed the Proportional Self-Organizing Map (PSOM) for clustering growers and finding the locations of loading stations by all data vectors being classified under both distance and capacity constraints. However, the result from the PSOM may not be the final solution, since some low-utilization clusters which have an amount of sugar cane less than the allowed minimum station capacity should be pruned by each farm and be merged into the nearest cluster. Recently, Boonmee et al. (2013) applied one of clustering techniques called Growing Neural Gas (GNG) to solve problems in allocation and transportation of pullets to hen houses with the aim of minimizing the cost of houses used, hen transportation costs, and loss from mixing hens at different ages in the same hen house. However, in terms of transportation costs reported in their work the improvements were limited (−0.05 to 4.11 %) when compared with the improvement in cost of houses used (3.62–22.60 %) and with loss from mixing cost (24.84–58.23 %). It means the phase of routing should be further developed in future research in order to improve the efficiency of hen transportation.

Since, in our problem, the distance information alone is not sufficient for determining the clusters of pullet houses, we need to consider several attributes such as the capacity of each house. Therefore, both distance and capacity constraints such as in the research of PSOM were considered in order to find a more effective solution. For these reasons, a new architecture of clustering algorithms called the Multifactor Based Evolving Self-Organizing Maps algorithm (MESOM) was developed based on the traditional Evolving-SOM algorithm by Dechampai et al. (2013). The MESOM is used to cluster pullet houses in order to minimize traveling distance. Afterwards, DE is applied first to determine routes of transferring pullets to hen houses which is a special case of VRP with consideration of flexibility of mixing pickup and delivery services and the maximum duration of a route constraint and split deliveries where pullets can be transferred to a hen house by more than one vehicle. It is the main contribution of this paper. Details of the algorithms will be described in the next section.

Methodology

Two heuristics are proposed to solve the problem. The first heuristic is called Differential Evolution for G-Q-DVRP with consideration of flexibility of mixing pickup services, delivery services and a maximum duration of a route constraint (DE_G-Q-DVRP-FD) and it directly applies the differential evolution technique (DE), while the second heuristic called Multifactor Based Evolving Self-Organizing Maps with Differential Evolution for G-Q-DVRP with consideration of flexibility of mixing pickup and delivery services and a maximum duration of a route constraint (MESOMDE_G-Q-DVRP-FD) which is a two-phase algorithm. The first phase of the MESOMDE_G-Q-DVRP-FD is to cluster the pullet houses. In this phase, the MESOM technique was used by employing both the distance factor and the capacity factor for solving the transportation problem. Furthermore, the pullet capacity of each house can be split into two nodes in order to completely utilize the capacity of each vehicle, and to reduce the steps of pruning after clustering. Then the routing for transferring pullets from the pullet houses to the hen houses is determined in the second phase with the objective to minimize the total transportation costs under the restrictions of capacity, a maximum duration of a route, and multiple depots. The differential evolution as detailed in the first algorithm (DE_G-Q-DVRP-FD) is also applied in this phase. Details of the proposed algorithms are as follows.

DE_G-Q-DVRP-FD

Differential evolution (DE) is the stochastic, population-based optimization algorithm that firstly introduced by Storn and Price in 1997. Such methods are commonly known as meta-heuristics that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. The candidate solution is presented as an initial vector population for each generation utilizes *NP*, *D*-dimensional parameter vectors $X_{i,G}$; $i = 1, 2, \ldots, NP$. In the first step, the initial vector population is chosen randomly and mostly generated in a uniform probability distribution. The initial vector population is evaluated the cost function of the target vector. Next, DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector in the mutation operation. The mutated vector's parameters are then mixed with the parameters of another predetermined vector, the target vector, to yield the so-called trial vector. Finally, in selection process, the trial vector and the target vector are compared, if the trial vector yields a lower cost function value than the target vector, the trial vector replaces the target vector in the following generation. Each population vector has to serve once as the target vector so that *NP* competitions take place in one generation (Storn and Price 1997). The process is repeated until there is no improvement of the objective function.

Since the traditional DE solves the optimization problem, it has obtained prestige as a very effective global optimizer. In this section, the new algorithm called "Differential Evolution for G-Q-DVRP with consideration of the flexibility of mixing pickup and delivery services and the maximum duration of a route constraint (DE_G-Q-DVRP-FD)" is developed to enhance company management in pullet transportation. In order to appropriately apply to real problems, this technique considers capacity restriction, maximum duration of a route, and multiple depots with the objective to minimize total costs.

Prior to the presentation of the algorithms, the indices and parameters used in modeling the problem are defined as follows:

Indices

i Index for pullet houses; $i = 1, 2, \ldots, I$
j Index for hen houses; $j = 1, 2, \ldots, J$
k Index for number of vehicles k; $k = 1, 2, \ldots, K$
a Priority order of ROV for pullet allocation; $a = 1, 2, \ldots, a_{\max}$
b Priority order of ROV for hen allocation; $b = 1, 2, \ldots, b_{\max}$

Parameters

d_{ij} Distance from pullet house i to hen house j [km]
C^F Fuel cost [unit per kilometer]
C^T Rent for vehicle k [unit per vehicle]
X_i^a Pullet house i, priority order no. a
Y_j^b Hen house j, priority order no. b
T_k Distance of vehicle no. k
$T_{a,b}$ Distance between pullet house order no. a to hen house order no. b
$T_{b,a}$ Distance between hen house order no. b to pullet house order no. a
T_{max} Maximum duration of a route of each vehicle
$cap.X_i^a$ Capacity of a pullet house i, priority order no. a
$cap.Y_j^b$ Capacity of a hen house j, priority order no. b
$maxcap.truck$ Maximum capacity of a vehicle
$cap.truck_k$ Capacity of a vehicle no. k

Decision variables

p^k Amount of pullets to be transported by vehicle no. k
r^k Solution set of pullet and hen quantities to be transported by vehicle no. k
s^k Solution set of house routing sequence (pickup from pullet house and delivery to hen house of vehicle no. k)
X_{ijk} $= 1$; when pullet house i is transported to hen house j, vehicle number k
$= 0$; otherwise
Q_k Number of vehicles k used to transport the commodity (Integer value)

Moreover, the problem considered is very complex in terms of house scatter, sizes of vehicles, and transportation time, some parts of DE algorithm are modified in order to determine cost function and suitably applied for allocation pullets to hen houses. Therefore, after the crossover process, the allocation of pullets to hen houses is added. Finally, Selection process is conducted. Overall step of the DE_G-Q-DVRP-FD is presented in Fig. 3.

According to the DE_G-Q-DVRP-FD procedure, there are four operations: Solution initialization, mutation, recombination, and selection operations. Detail of each operation is presented in the following:

Step 1: Solution initialization

Generating the initial vector population is an important part of DE and a good initial population directly affects the solution to reach better quality. The design of the popula-

Fig. 3 Steps of the proposed method (DE_G-Q-DVRP-FD)

```
Procedure of DE_G-Q-DVRP-FD algorithm
Begin Encoding step: randomly generate a set of target vectors $n$ ($n$ =1…$NP$);
    // $n$ is index of the vector representation; $NP$ is predefined number of population
    while termination condition is not satisfied do
        for $n$ = 1 to $NP$            //$NP$ is predefined number of population
            Solution Initialization
                Decoding and Evaluation
            Mutation Operation
            Crossover Operation
            Selection Operation
        endfor
    endwhile
end;
```

tion and its parameters is represented in the encoding and decoding procedures shown in the following:

(1) Encoding method

In DE, an initial vector population is a set of parameters which define a proposed solution to the problem that is trying to solve. Firstly, in the encoding step, the initial vector population is randomly generated in the uniform number between [0,1) of both target vectors $X_{n,G}$ and $Y_{n,G}$ for $n = 1, \ldots, NP$, where NP is a number of population for each generation G. The initial vector population in the problem is separated in two parts. The first part is necessarily specific to pullet house routing problem in the form of vector $Unclassified.X = \left\{X_i \in R^2 | i = 1, .., I\right\}$ and the second part is specific to hen house routing problem in the form of vector $Unclassified.Y = \left\{Y_j \in R^2 | j = 1, .., J\right\}$. Two vectors are the sets of pullet house and hen house locations which are defined as the 2-D vectors relying on XY-coordinate system, respectively as show in Fig. 4. The real number shown in each position of the vectors in the first iteration of the proposed algorithm obtains from randomly generate while the next iteration the position values gain from the differential evolution algorithm operators (mutation and recombination process).

(2) Decoding method

From Fig. 4, the position value of both X and Y vector is sorted. The sorting of the pullet and hen house index bases on its corresponding position value. According to the priority order of value (ROV) rule, the smaller position value is defined to the higher priority of the pullet and hen house allocation (see Fig. 5).

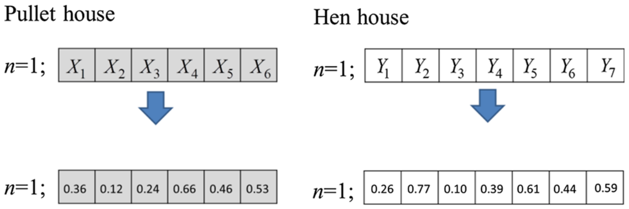


Fig. 4 Encoding of the initial vector population $n = 1$ for pullet and hen house routing problem

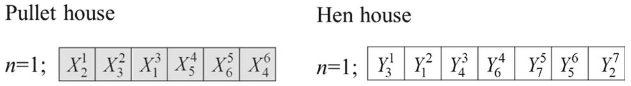

Pullet house
Hen house
n=1;
X_2^1 X_3^2 X_1^3 X_5^4 X_6^5 X_4^6
n=1;
Y_3^1 Y_1^2 Y_4^3 Y_6^4 Y_7^5 Y_5^6 Y_2^7

Fig. 5 ROV of the initial vector population $n = 1$ for pullet and hen house routing problem

The following conceptual framework of the allocation process is shown in Fig. 6. Allocation of pullets to hen houses can be done by ranking the number of ROV values. All pullets in the houses must be transferred ($Unclassified. = \emptyset$) to hen houses which have enough capacity. In transporting pullets, only one vehicle is considered at a time until all pullets are completely transported. The ROV ranking is used to consider how many of pullets to be transported. If the vehicle can transport pullets as its maximum capacity from the first pullet house (X_i^a) to deliver them to the first hen house (Y_j^b), the vehicle will pick up and deliver only one time each. In order to obtain high vehicle utilization, a vehicle can pick up pullets from more than one houses, and also deliver them to more than one hen houses unless the $maxcap.truck$ and $Tmax$ constraints are met. Each vehicle will have its own sequences (s^k) and amount of pullets to pick up and/or delivery (r^k).

Once the pullets from the first pullet house are completely picked up ($cap.X_i^a == 0$), the next pullet house is considered ($X_i^a == X_i^{a+1}$). Using the same concept, delivery pullets to the first hen houses must be completed ($cap.Y_j^b == 0$) before delivering to the next hen house ($Y_j^b == Y_j^{b+1}$). Once the pullets from the pullet house are picked up and delivered to the hen house, the sets of pullet houses ($Unclassified.X \leftarrow Unclassified.X - X_i^a$) and hen houses ($Unclassified.X \leftarrow Unclassified.X - Y_j^b$) for picking up and delivering for the next round are updated by deleting the houses already picked up and delivered.

```
Procedure: Allocation operation
begin
$k \leftarrow 1$
    while $Unclassified.X \neq \varnothing$ do
        $s^k \leftarrow s^k \cup X_i^a$                              //update sequence of route
      while $cap.truck_k > 0$ or $T_k \geq T_{max}$ do
          $p^k \leftarrow \min\{cap.truck_k, \min\{cap.X_i^a, cap.Y_j^b\}\}$
          $r^k \leftarrow r^k + p^k$                              //update quantity of pullet transporting
          $s^k \leftarrow s^k \cup Y_j^b$                          //update sequence of route
          $cap.truck_k \leftarrow cap.truck_k - p^k$              //update remaining capacity of a vehicle k
             if $cap.truck_k == 0$ then
               $s^k \leftarrow s^k \cup X_i^a$                    //update sequence of route
               $T_k \leftarrow T_k + T_{b,a}$                     //update remaining route duration of vehicle k
             endif
          $cap.X_i^a \leftarrow cap.X_i^a - p^k$                  //update remaining quantity of pullet house a
          $cap.Y_j^b \leftarrow cap.Y_j^b - p^k$                  //update remaining quantity of hen house b
          $T_k \leftarrow T_k + T_{a,b}$                          //update remaining route duration of vehicle k
          if $cap.X_i^a == 0$ then
            if $X_i^a == X_i^{a\max}$ then
               $T_k = T_{max}$, $Unclassified.X = \varnothing$
            endif
            $X_i^a == X_i^{a+1}$
            $s^k \leftarrow s^k \cup x_i^a$                       //update sequence of route
            $Unclassified\ X \leftarrow Unclassified\ X - X_i^a$  //update the set of unclassified pullet houses
            $T_k \leftarrow T_k + T_{b,a}$                        //update remaining route duration of vehicle k
          endif
          if $cap.Y_j^b == 0$ then
            $Y_j^b == Y_j^{b+1}$
            $Unclassified.X \leftarrow Unclassified.X - Y_j^b$ //update the set of unclassified hen houses
          endif
      endwhile
      $cap.truck_k = cap.truck$
    endwhile
$k \leftarrow k + 1$
end;
```

Fig. 6 Procedure for allocating pullets to hen houses

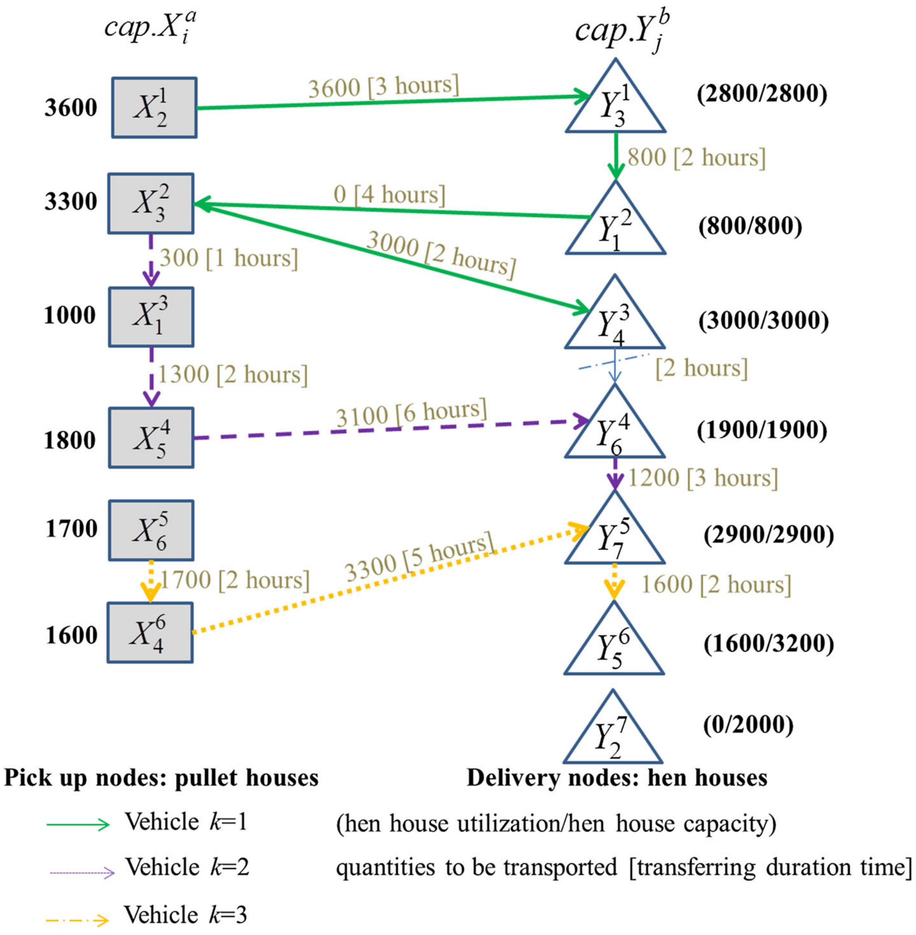
These pullet and hen houses already picked up and delivered will be stored in the sets of vehicles $s^k \leftarrow s^k \cup X_i^a$ and $s^k \leftarrow s^k \cup Y_j^b$, respectively. The amount of pullets already transported to hen houses will be stored in the set r^k and updated as $r^k \leftarrow r^k + p^k$.

Example of decoding

The sorted list of pullet and hen house of the initial vector population in Fig. 5 is considered as the priority list to assign routing between the pullet house and hen house. The maximum capacity of vehicle is given as $maxcap.truck = 3600$ hens per vehicle and the maximum duration of a route $(Tmax)$ is 12 h.

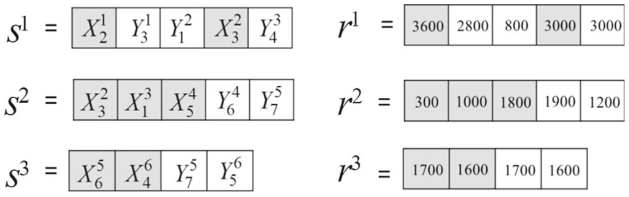
Under these two constraints, it can be seen from Fig. 7 that the Vehicle no. 1 picks up 3600 pullets from pullet house $i = 2$ and deliver 2800 and 800 pullets to hen houses $j = 3$ and $j = 1$, respectively. Total time duration for transferring from pullet house $i = 2$ to hen houses $j = 3$ and $j = 1$ are five hours $(3 + 2 = 5\,\text{h})$. Therefore, we can see that Vehicle no. 1 still has remaining time $(12 - 5 = 7\,\text{h})$ to pick up 3000 pullets from house $i = 3$ and delivers all of them to hen house $j = 4$ that requires six hours $(4 + 2 = 6\,\text{h})$ for transferring. We can see that Vehicle no. 1 still has 300 pullets as remaining capacity to pick up. However, the vehicle has only one hour remaining $(7 - 6 = 1\,\text{h})$ to pick up the pullets from this route. Hence, Vehicle no. 1 cannot deliver these pullets to

Fig. 7 Decoding of transferring pullets to hen houses obtained of solution set $n = 1$



the next priority hen houses ($j = 6$), since the time duration for travelling from hen house $i = 4$ to hen houses $j = 6$ takes two hours violating to the $Tmax$ constraint. Therefore, the second vehicle ($k \leftarrow k+1$) is considered to pick up these 300 pullets from pullet house $i = 3$ and also pullet houses $i = 1$ and $i = 5$ for 300, 1000, and 1800 pullets, respectively. These pullets will then be transferred to hen houses $j = 6$ and $j = 7$, for 1900 and 1200 pullets, respectively. The process is repeated until all pullets are transferred to hen houses ($Unclassified. = \emptyset$).

Subsequently, the solution set is obtained from decoding which is the sequence for transferring pullets to hen houses, and also the quantity of pullets transported by each vehicle as shown in Fig. 8.



$S^1 = X_2^1 \; Y_3^1 \; Y_1^2 \; X_3^2 \; Y_4^3$ $r^1 = 3600 \; 2800 \; 800 \; 3000 \; 3000$

$S^2 = X_3^2 \; X_1^3 \; X_5^4 \; Y_6^4 \; Y_7^5$ $r^2 = 300 \; 1000 \; 1800 \; 1900 \; 1200$

$S^3 = X_6^5 \; X_4^6 \; Y_7^5 \; Y_5^6$ $r^3 = 1700 \; 1600 \; 1700 \; 1600$

Fig. 8 Solutions obtained from decoding for transferring pullets to hen houses of solution set $n = 1$

When all pullets are completely allocated to hen houses and the routes of pullet transportation are assigned, the total cost (TC) as objective function for each of n parameter vectors on iteration t is calculated in terms of fuel cost ($fuel\_cost$) and vehicle renting cost (veh_rent) as shown in Eq. (1).

TC Total cost including fuel cost ($fuel\_cost$) and vehicle renting cost (veh_rent)

$$= fuel\_cost + veh\_rent$$
$$= \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(C^F \times d_{ij} \times X_{ijk}) + (C^T \times Q_k) \qquad (1)$$

Step 2: Mutation operation

Mutation is performed in order to obtain new solutions differently from the initial population. Mutation can be done by taking random vector to find the difference in the following process.

(1) Define Target vectors $X_{n,G}$ and $Y_{n,G}$, where n = 1, 2, 3,…,NP

(2) Random 3 vectors from the initial population differently from the target vector (i.e., $n \neq r1 \neq r2 \neq r3$). The

random vectors are $X_{r1,G}, X_{r2,G}, X_{r3,G}$ for pullet houses, and $Y_{r1,G}, Y_{r2,G}, Y_{r3,G}$ for hen houses.

(3) Calculate the mutant vector from the equations: $Vx_{n,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$ for pullet houses and $Vy_{n,G+1} = Y_{r1,G} + F(Y_{r2,G} - Y_{r3,G})$ for henhouses. The mutation factor (F), which is a real constant used is 2.

Step 3: Recombination operation

In the recombination operation, each of the mutant vectors does a crossover with the target vector to generate the trial vectors $Ux_{i,n,G+1}$ and $Uy_{j,n,G+1}$ for pullet house and hen house, respectively.*CR* is the crossover rate ∈ [0, 1]. After the experiment, the *CR* is chosen as 0.80 appropriately for the problem. The trial vectors are formed by the following equations:

$$Ux_{i,n,G+1} = \begin{cases} Vx_{i,n,G+1} & \text{if } rand_i \le CR \\ X_{i,n,G} & \text{otherwise} \end{cases}$$

$$Uy_{j,n,G+1} = \begin{cases} Vy_{j,n,G+1} & \text{if } rand_j \le CR \\ Y_{j,n,G} & \text{otherwise} \end{cases}$$

The recombination operation is carried out by:

(1) generating random numbers, $rand_i$ for pullet houses and $rand_j$ for hen houses, within interval [0,1],

(2) for each of *NP*: if $rand_i \le CR$ and $rand_j \le CR$; copy the value from the mutant vector, else copy the value from the target vector into the trial vector (see Fig. 9).

(3) Then, update the values of trial vector in range between [0, 1] using the following equations.

$$Ux^{*}_{i,n,G} = \frac{MaxTarget - MinTarget}{MaxU_n - MinU_n} \times (Ux_{i,n,G} - MinU_n),$$

$$Uy^{*}_{j,n,G} = \frac{MaxTarget - MinTarget}{MaxU_n - MinU_n} \times (Uy_{j,n,G} - MinU_n)$$

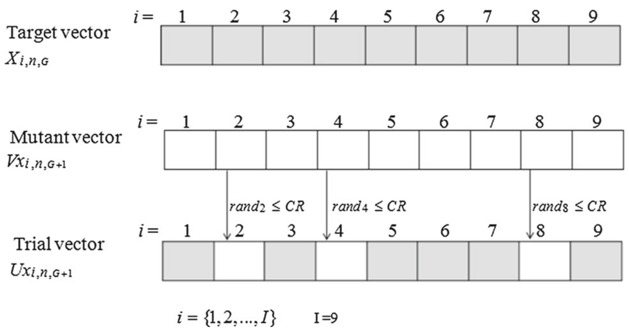


Fig. 9 Illustration of recombination operation of pullet house

(4) Finally, sorting the values of trial vector in ascending order (or rank order value: ROV).

Step 4: Selection operation

The last operation is called selection that to decide whether or not it should become the initial solution of the next generation $(G + 1)$ using the format of one-to-one competition scheme to select new candidates greedily. The objective function of the trial vectorsare compared with that of the target vectors, the vector with the lowest value of the two becomes "Individual 1" for the next generation, according to the following equation:

$$X_{n,G+1} = \begin{cases} Ux_{n,G+1} & \text{if } f(U_{n,G+1}) \le f(X_{n,G}) \\ X_{n,G} & \text{otherwise} \end{cases}$$

$$Y_{n,G+1} = \begin{cases} Uy_{n,G+1} & \text{if } f(U_{n,G+1}) \le f(X_{n,G}) \\ Y_{n,G} & \text{otherwise} \end{cases}$$

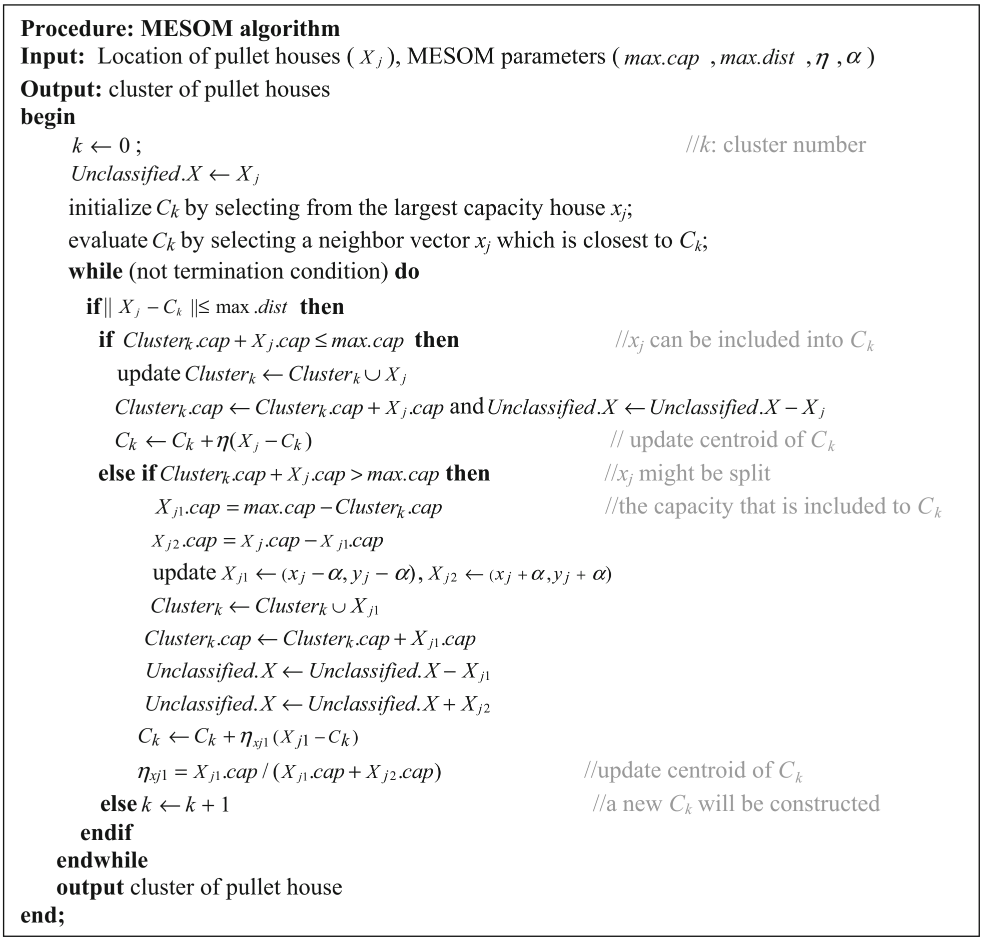
To evolve "Individual 2" for the next generation, the second member of the population is set as target vector and the above process is repeated. This process is repeated *NP* times until the new population set array is filled which completes one generation. Once the termination criterion is met, the algorithm ends. If the trial vector yields a lower cost function value than the target vector, the trial vector replaces the target vector in the following generation.

MESOMDE_G-Q-DVRP-FD

Since the problem considered in this paper is very complex, the new algorithm called a "Multifactor Based Evolving Self-Organizing Maps with Differential Evolution for G-Q-DVRP with consideration of flexibility of mixing pickup and delivery services and a maximum duration of a route constraints (MESOMDE_G-Q-DVRP-FD)" was developed to improve the solution of the problem. Details of the two-phase algorithm are as follows.

Phase 1: Clustering phase

According to Dechampai et al. (2013), the algorithm called "Multifactor Based Evolving Self-Organizing Maps (MESOM)" was developed to cluster pickup customer vertices (i.e., pullet houses) in order to reduce transportation cost. Due to its effectiveness, the MESOM is therefore used in this study. The MESOM was developed based upon the traditional Evolving Self-organizing Maps algorithm (ESOM). The modifications are that, in the MESOM, the winner node was adjusted and the calculation of the neighborhood function was ignored to determine the online K-mean. Resulting from each cluster has a centroid which does not depend on one cluster, the cluster updates itself until it is fulfilled, creating a new cluster (node) when necessary. Therefore, this



```
Procedure: MESOM algorithm
Input: Location of pullet houses ( $X_j$ ), MESOM parameters ( $max.cap$ , $max.dist$ , $\eta$ , $\alpha$ )
Output: cluster of pullet houses
begin
        $k \leftarrow 0$ ;                                                    //$k$: cluster number
        $Unclassified.X \leftarrow X_j$
        initialize $C_k$ by selecting from the largest capacity house $x_j$;
        evaluate $C_k$ by selecting a neighbor vector $x_j$ which is closest to $C_k$;
        while (not termination condition) do
          if $\| X_j - C_k \| \le \max.dist$ then
            if $Cluster_k.cap + X_j.cap \le max.cap$ then               //$x_j$ can be included into $C_k$
              update $Cluster_k \leftarrow Cluster_k \cup X_j$
              $Cluster_k.cap \leftarrow Cluster_k.cap + X_j.cap$ and $Unclassified.X \leftarrow Unclassified.X - X_j$
              $C_k \leftarrow C_k + \eta(X_j - C_k)$                      // update centroid of $C_k$
            else if $Cluster_k.cap + X_j.cap > max.cap$ then            //$x_j$ might be split
                $X_{j1}.cap = max.cap - Cluster_k.cap$                  //the capacity that is included to $C_k$
                $X_{j2}.cap = X_j.cap - X_{j1}.cap$
                update $X_{j1} \leftarrow (x_j - \alpha, y_j - \alpha)$, $X_{j2} \leftarrow (x_j + \alpha, y_j + \alpha)$
                $Cluster_k \leftarrow Cluster_k \cup X_{j1}$
                $Cluster_k.cap \leftarrow Cluster_k.cap + X_{j1}.cap$
                $Unclassified.X \leftarrow Unclassified.X - X_{j1}$
                $Unclassified.X \leftarrow Unclassified.X + X_{j2}$
                $C_k \leftarrow C_k + \eta_{xj1}(X_{j1} - C_k)$
                $\eta_{xj1} = X_{j1}.cap / (X_{j1}.cap + X_{j2}.cap)$   //update centroid of $C_k$
            else $k \leftarrow k + 1$                                   //a new $C_k$ will be constructed
          endif
      endwhile
      output cluster of pullet house
end;
```

Fig. 10 The procedure of the MESOM algorithm (Dechampai et al. 2013)

method does not need to consider weak connections or calculate connection strength. Each house will become a data vector in a 2-D space, after mapping the pullet house locations onto the XY planar axis. The MESOM network starts clustering at the largest capacity house. Each cluster only represents one vehicle. Any house in a cluster should send pullets to the vehicle of the corresponding cluster. During learning, the network updates itself and creates a new cluster when it exceeds the maximum distance or inordinate vehicle capacity constraints. In this algorithm, two sub phases are developed. Firstly, the splitting method and then multifactor are used to cluster the pullet houses with splitting-coefficient of centroid movement (η_{xij}) in order to completely utilize the transportation.

Since the classical ESOM algorithm uses only the distance criterion for constructing the topology, in regards to our problem, only distance information is not sufficient for determining the clusters of houses. Therefore, Dechampai et al. (2013) apply the algorithm including additional concerns with the qualification of several attributes such as the capacity of each vehicle and the transportation distance criteria. These qualifications are to play an essential role for clustering the houses in order to be appropriate for application in real problems. Therefore their method was eventually known as Multifactor in the MESOM procedure.

The MESOM algorithm can be explained step by step as shown in Fig. 10.

Phase 2: Allocation and routing phase

According to the MESOMDE_G-Q-DVRP-FTD procedure, after the clusters of pullet houses (C_k) are determined from the first phase, the determination of routes is established in this phase. The procedure for determining vehicle routing is same as that of the DE_G-Q-DVRP-FTD algorithm presented in DE_G-Q-DVRP-FD section. When all pullets are completely allocated to hen houses and the routes of pullet transportation are assigned, Eq. (1) is also used to calculate the total cost (TC) or objective function for each of n parameter vectors on iteration t.

Table 2 Values of parameters used in each data set

Data set	Vehicle capacity (units per vehicle)	Total number of pullet houses (houses)	Total number of hen houses (houses)	Rate for vehicle renting (Baht/vehicle/day)[a]	Fuel cost (Baht/km)[a]
1	2300	20	60	2200	13.9
2	2300	20	80	2200	13.9
3	2300	30	60	2200	13.9
4	2300	30	80	2200	13.9
5	2300	40	60	2200	13.9
6	2300	40	80	2200	13.9
7	3600	20	60	2500	11.4
8	3600	20	80	2500	11.4
9	3600	30	60	2500	11.4
10	3600	30	80	2500	11.4
11	3600	40	60	2500	11.4
12	3600	40	80	2500	11.4

[a] One US dollar is approximately 30 Baht

Table 3 Comparison of the average number of vehicles used and the average traveling distance between the current practices and using the DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms for different data sets

Data type	Average number of vehicles used			Average travelling distance		
	Current practice	DE_G-Q-DVRP-FD	MESOMDE_G-Q-DVRP-FD	Current practice	DE_G-Q-DVRP-FD	MESOMDE_G-Q-DVRP-FD
1	31	30	28	2918	2817	2649
2	29	28	27	2908	2674	2640
3	39	38	35	4227	3590	3358
4	37	36	33	4035	3533	3284
5	50	47	42	5491	4496	4050
6	44	44	40	4311	4216	3841
7	29	26	22	2918	2527	2066
8	27	24	21	2908	2327	2033
9	36	35	29	4227	3354	2836
10	35	33	27	4039	3207	2706
11	48	44	35	5491	4263	3331
12	42	42	34	4311	4031	3279

Case study application

This research was motivated by the interest of an egg industry in northeastern Thailand. Currently, the daily production of the firm is about 1,200,000 eggs with the total number of laying hens in its houses about 1,500,000 hens. Presently, the chicks are ordered weekly at 30,000–50,000 chicks from hatcheries depending on egg demand volumes in the planning horizon. The chicks are then moved to the 21 contracted pullet-raising houses with different capacities ranging from 15,000 to 35,000 pullets per house. When the pullets are 17 weeks old, they are transferred to hen houses for laying eggs, and hens will remain in the hen house until the age of 75 weeks for egg production. The hen houses are located within the radius of 50–100 km from the pullet houses. To transport the hens, the firm has a contract with a third-party logistics provider using only 6-wheel vehicles with a capacity of 2300 hens. Presently, there are 82 contracted hen houses with different capacities ranging from 5400 to 59,100 laying hens per house.

To allocate hens to hen houses, the firm currently considers only two major factors, namely, distance between the pullet houses and the hen houses and sizes of the hen houses. In order to minimize the travelling distance of the vehicle, the hen house with the shortest distance is given priority rather than the size of the hen houses. In other words, the hen house

Table 4 Computational results for each cost component including total cost for all data sets (thousands of Baht)

Data Set	Cost of vehicle utilization					Fuel cost					Total cost				
	Current practice	DE_G-Q-DVRP-FD		MESOMDE_G-Q-DVRP-FD		Current practice	DE_G-Q-DVRP-FD		MESOMDE_G-Q-DVRP-FD		Current practice	DE_G-Q-DVRP-FD		MESOMDE_G-Q-DVRP-FD	
		Best	Mean	Best	Mean		Best	Mean	Best	Mean		Best	Mean	Best	Mean
1	68.20	61.60	66.00	55.00	60.50	40.56	38.45	39.15	32.73	36.82	108.76	100.05	105.15	87.73	97.32
2	63.80	52.80	61.16	57.20	59.62	40.42	34.82	37.16	35.74	36.70	104.22	87.62	98.32	92.94	96.32
3	85.80	77.00	83.16	70.40	77.22	58.76	43.45	49.90	44.22	46.67	144.56	120.45	133.06	114.62	123.89
4	81.40	74.80	78.98	66.00	71.94	56.09	46.61	49.11	42.56	45.65	137.49	121.41	128.09	108.56	117.59
5	110.00	96.80	102.96	83.60	92.18	76.32	58.46	62.50	52.39	56.30	186.32	155.26	165.46	135.99	148.48
6	96.80	90.20	96.80	81.40	87.12	59.92	53.78	58.61	50.25	53.39	156.72	143.98	155.41	131.65	140.51
7	72.50	60.00	64.50	52.50	54.00	33.27	28.74	30.07	23.06	24.58	105.77	88.74	94.57	75.56	78.58
8	67.50	57.50	60.00	47.50	53.50	33.15	26.17	27.69	22.46	24.19	100.65	83.67	87.69	69.96	77.69
9	90.00	85.00	87.25	70.00	73.25	48.19	37.83	39.91	31.27	33.75	138.19	122.83	127.16	101.27	107.00
10	87.50	80.00	82.75	60.00	68.00	46.04	36.32	38.16	30.20	32.20	133.54	116.32	120.91	90.20	100.20
11	120.00	100.00	110.00	80.00	87.50	62.60	47.53	50.73	37.66	39.64	182.60	147.53	160.73	117.66	127.14
12	105.00	95.00	104.75	77.50	84.25	49.15	44.66	47.97	36.06	39.02	154.15	139.66	152.72	113.56	123.27

Best = the best solution found among 10 runs for each data set
Mean = the average value on 10 runs for each data set

with the shortest distance between the pullet house and the hen house is chosen first. If there are more hen houses with the same distance, the hen house will be selected in the order of larger size. In the case where a hen house is only partially filled, that hen house may be used in later weeks. The hen house with the remaining capacity will be selected based on the prescribed criterion. However, the simple practice of the firm may cause an inefficient allocation in the long run. It may have some inherent disadvantages, especially an increase of transportation distance charged at 13.9 units per kilometer. It should be noted here that monetary unit in our paper is in baht (assuming one US dollar is approximately 30 baht).

Numerical experiments

Since the case of this practical problem is very complex, it cannot be solved by a simple calculation. For this reason, the integration of a comprehensive decision support system (DSS) is required. To solve the problem, the DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms were developed using Matlab version 7.9.0.529 on AMD E-450 APU with Radeon$^{\text{TM}}$ HD Graphics (1.65 GHz), with 4.00 GBytes of RAM. In order to test the model, recorded data of hen allocation of the company during the year 2010–2012 was used to test the performance of the proposed DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD methods in terms of total costs, consisting of fuel cost and vehicle renting cost. Twelve sets of problems were generated based on three types of data characteristics of the current practice in order to evaluate the performance of the heuristics algorithms. The three types of data characteristics were sizes of vehicles, total number of pullet houses, and total number of hen houses. In the real case, only one size of vehicle (i.e., 6-wheel vehicle with 2300 units of capacity) is used. However, a 10-wheel vehicle is another size of vehicle normally used in transporting commodities and will be planned for use by the company in the near future. Hence, in this study, two levels of vehicle sizes were considered, low (i.e., 2300 pullets per vehicle) and high (i.e., 3600 pullets per vehicle) to see if the 10-wheel vehicle is more economical. For the two other parameters, three levels of total number of pullet houses, low (i.e., 20 pullet houses), medium (i.e., 30 pullet houses), and high (i.e., 40 pullet houses), and two levels of total number of hen houses, low (i.e., 60 hen houses) and high (i.e., 80 hen houses) are considered. For each combination, ten test problems were generated. The parameters for each combination are given in Table 2.

To investigate the performance of the proposed DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms, the relative improvement (*RI*) of the solutions obtained by the algorithms with respect to those of the current practice was determined by Eq. (2).

Table 5 Average relative improvement results (%) from the DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms for different data sets

Data set	Cost of vehicle utilization		Fuel cost		Total cost	
	DE_G-Q-DVRP-FD	MESOMDE_G-Q-DVRP-FD	DE_G-Q-DVRP-FD	MESOMDE_G-Q-DVRP-FD	DE_G-Q-DVRP-FD	MESOMDE_G-Q-DVRP-FD
1	3.23	11.29	3.46	9.23	3.31	10.52
2	4.14	6.55	8.06	9.22	5.66	7.59
3	3.08	10.00	15.08	20.57	7.95	14.30
4	2.97	11.62	12.44	18.61	6.84	14.47
5	6.40	16.20	18.11	26.24	11.20	20.31
6	0.00	10.00	2.20	10.91	0.84	10.35
7	11.03	25.52	13.39	29.20	11.78	26.68
8	11.11	20.74	19.98	30.10	14.03	23.82
9	3.06	18.61	20.66	32.91	9.19	23.60
10	5.43	22.29	20.61	33.00	10.66	25.98
11	8.33	27.08	22.37	39.33	13.15	31.28
12	0.24	19.76	6.49	23.94	2.23	21.10

Table 6 Comparison of the unit cost (unit/pullet) between the current practice and using the DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms for different types of vehicle

Data set	Current practice	DE_G-Q-DVRP-FD	MESOMDE_G-Q-DVRP-FD
6-wheel vehicle	1.80	1.69	1.57
10-wheel vehicle	n/a[a]	1.57	1.30

[a] Currently, the case study company has not used this size of vehicle yet

Let

$$RI = ((S_{cur} - S_{heu})/S_{cur}) \times 100 \tag{2}$$

where RI = the relative improvement (%) between S_{cur} and S_{heu}. S_{cur} = the solution obtained from the current practice. S_{heu} = the solution obtained from the proposed algorithms (DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD).

The solution of each test problem representing the current practice of the company and that using the proposed DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms including the relative improvement were obtained for all test problems of each combination. To demonstrate the efficiency of the MESOMDE_G-Q-DVRP-FD algorithm on these two cost components, Table 3 shows a comparison of the results of the current practice and those of the DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms for each data set. We can see that the MESOMDE_G-Q-DVRP-FD algorithm yields the lowest number of vehicles used and the lowest travel distances. Table 4 shows the best and the average values of each cost component including the total cost obtained on 10 runs for each data set. Based on these results, the relative improvement values obtained for all test problems of each data set are presented in Table 5. It can be seen that the relative improvement of the total cost ranged from 7.59 to 31.28 %, with an average of 19.17 % for the MESOMDE_G-Q-DVRP-FD algorithm, and 0.84–13.15 %, with an average of 8.07 % for the DE_G-Q-DVRP-FD algorithm. Also, from this table, it can be seen that both of the proposed algorithms can reduce costs greatly, especially the fuel cost. However, the proposed MESOMDE_G-Q-DVRP-FD algorithm gave a

Table 7 Paired sample test of the total cost between DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms

Pair 1	Paired differences					t	*df*	Sig. (2-tailed)
	Mean	Std. deviation	Std. error mean	95 % confidence interval of the difference				
				Lower	Upper			
Paired samples test								
DE_G-Q-DVRP-FD - MESOMDE_G-Q-DVRP-FD	15796.25000	8934.54076	2579.17976	10119.51363	21472.98637	6.125	11	.000

Table 8 Paired sample test of the total distance between DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms

Pair 2	Paired differences					t	*df*	Sig. (2-tailed)
	Mean	Std. deviation	Std. error mean	95 % confidence interval of the difference				
				Lower	Upper			
Paired samples test								
DE_G-Q-DVRP-FD - MESOMDE_G-Q-DVRP-FD	413.49167	249.59337	72.05140	254.90761	572.07573	5.739	11	.000

Table 9 Paired sample test of the number of vehicles between DE_G-Q-DVRP-FD and MESOMDE_G-Q-DVRP-FD algorithms

Pair 3	Paired differences					t	*df*	Sig. (2-tailed)
	Mean	Std. deviation	Std. error mean	95 % confidence interval of the difference				
				Lower	Upper			
Paired samples test								
DE_G-Q-DVRP-FD - MESOMDE_G-Q-DVRP-FD	4.49167	2.41866	0.69821	2.95492	6.02841	6.433	11	.000

Table 10 Computational results for each cost component including total cost for all data sets (thousands of Baht)*

Data set	Cost of vehicle utilization				Fuel cost				Total cost			
	GNG		MESOMDE_G-Q-DVRP-FD		GNG		MESOMDE_G-Q-DVRP-FD		GNG		MESOMDE_G-Q-DVRP-FD	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean
1	88.00	88.00	55.00	60.50	23.53	24.32	32.73	36.82	111.53	112.32	87.73	97.32
2	88.00	88.00	57.20	59.62	27.35	28.78	35.74	36.70	115.35	116.78	92.94	96.32
3	110.00	110.00	70.40	77.22	25.54	29.48	44.22	46.67	135.54	139.48	114.62	123.89
4	110.00	110.00	66.00	71.94	23.71	24.80	42.56	45.65	133.71	134.80	108.56	117.59
5	121.00	121.00	83.60	92.18	45.67	49.37	52.39	56.30	166.67	170.37	135.99	148.48
6	121.00	121.00	81.40	87.12	26.26	27.55	50.25	53.39	147.26	148.55	131.65	140.51
7	130.00	130.00	70.00	73.33	34.11	34.87	32.08	32.43	164.11	164.87	102.08	105.77
8	130.00	130.00	67.50	70.00	32.69	34.27	30.91	31.65	162.69	164.27	98.41	101.65
9	162.50	162.50	87.50	87.50	28.19	29.56	39.46	40.63	190.69	192.06	126.96	128.13
10	162.50	162.50	82.50	85.00	25.15	26.12	37.73	38.44	187.65	188.62	120.23	123.44
11	217.50	217.50	137.50	138.33	48.43	50.55	59.72	60.88	265.93	268.05	197.22	199.22
12	217.50	217.50	127.50	127.50	34.49	36.81	55.03	56.21	251.99	254.31	182.53	183.71

Best = the best solution found among 10 runs for each data set
Mean = the average value on 10 runs for each data set

significantly lower cost for all cost components than the DE algorithm (see paired samples tests in Tables 7, 8 and 9) since the MESOMDE_G-Q-DVRP-FD was developed by considering both distance as the main factor and also the hen house capacity in order to cluster pullet houses as pickup customers and hen houses as delivery customers to reduce the travel distance, while the maximum duration of a route constraintexists. Once the pullet houses and hen houses are clustered, the differential evolution technique (DE) was developed for determining routes for transferring pullets from the pullet houses to the hen houses.

To demonstrate if the 10 wheel vehicle is more economical, Table 6 shows a comparison of cost per pullet transported between the two sizes of vehicles. We can see that the

Table 11 The difference (%gap) between the GNG and MESOMDE_G-Q-DVRP-FD algorithms of different data sets

Data set	%gap		
	Cost of vehicle utilization	Fuel cost	$Totalcost$
1	45.45	−33.95	15.41
2	47.60	−21.58	21.24
3	42.45	−36.83	12.58
4	52.91	−45.67	14.64
5	31.26	−12.31	14.74
6	38.89	−48.40	5.72
7	77.28	7.52	55.88
8	85.71	8.28	61.60
9	85.71	−27.25	49.89
10	91.18	−32.05	52.80
11	57.23	−16.97	34.55
12	70.59	−34.51	38.43

10-wheel vehicle yields the lower cost per unit transported for both proposed algorithms, and the MESOMDE_G-Q-DVRP-FD gives the lowest cost per unit transported.

Moreover, to investigate the efficiency of the MESOMDE_G-Q-DVRP-FD, it was compared with the benchmark problem reported in the literature. The method is Growing Neural Gas (GNG) proposed by Boonmee et al. (2013) (See "Appendix"). The GNG was developed to transfer pullets to hen houses by determining the centroid of hen houses before determining the routes. The objective of their study is to minimize the total cost consisting of house renting from farmer, hen transportation costs, and losses from mixing hens at different age. In order to obtain results that are comparable to the GNG algorithm and to represent the problem, we generate 6 new data types to replace to data sets 7–12. The comparison of the results of the GNG and those of the MESOMDE_G-Q-DVRP-FD algorithms for each data set shows in Table 10. In this table, the best and the average values of each cost component including the total cost obtained on 10 runs for each data set were presented. Based on these results, the difference (%*gap*) between two algorithms determined by Eq. (3) was obtained for all test problems of each data set and presented in Table 11.

Let

$$\%gap = ((S_{GNG} - S_{heu})/S_{heu}) \times 100 \tag{3}$$

where %*gap* the percentage of the difference between S_{GNG} and S_{heu}. S_{GNG} the solution obtained from the GNG algorithm. S_{heu} the solution obtained from the proposed algorithm (MESOMDE_G-Q-DVRP-FD).

From Tables 10 and 11, it can be seen that the MESOMDE_G-Q-DVRP-FD algorithm yields a significantly lower total cost, especially vehicle utilization cost since the MESOMDE_G-Q-DVRP-FD algorithm was developed for both pickup and delivery services in the same route. However, in most instances, the vehicles may travel more distance resulting in high fuel cost (see Table 12). In contrast, the GNG algorithm gives lower fuel cost since the algorithm mainly focuses on the short distance rather than the number of vehicles used in the system. Furthermore, when considering the total costs, the MESOMDE_G-Q-DVRP-FD algorithm is more efficient compared to the GNG algorithm which gives lower total cost than the GNG algorithm ranging from 5.72 to 61.60 %, with an average of 31.46 %.

Table 12 Comparison of the average number of vehicles used and the average traveling distance between the GNG and MESOMDE_G-Q-DVRP-FD algorithms for different data sets

Data type	Average number of vehicles used			Average travelling distance		
	Current practice	GNG	MESOMDE_G-Q-DVRP-FD	Current practice	GNG	MESOMDE_G-Q-DVRP-FD
1	31	40	28	2918	1750	2649
2	29	40	27	2908	2071	2640
3	39	50	35	4227	2121	3358
4	37	50	33	4035	1784	3284
5	50	55	42	5491	3552	4050
6	44	55	40	4311	1982	3841
7	39	52	29	3220	2510	2845
8	40	52	28	3062	2466	2776
9	46	65	35	4014	2127	3563
10	48	65	34	3864	1880	3372
11	65	87	55	5595	3638	5340
12	68	87	51	5211	2649	4930

Conclusions and discussion

In this paper, we propose new methods to solve the problem of hen allocation to hen houses with the objective of minimizing the total cost consisting of two cost components, cost of vehicle utilization and fuel cost. Then, the problem is known as the G-Q-DVRP with consideration of flexibility of mixing pickup and delivery services and the maximum duration of a route constraint. To solve the problem, two heuristic algorithms were developed. The DE_G-Q-DVRP-FD algorithm uses the differential evolution technique (DE). Since the problem considered is very complicated, the DE_G-Q-DVRP-FD was modified as a two-phase algorithm called MESOMDE_G-Q-DVRP-FD. In the first phase, clustering of pullet houses was determined before allocating pullets to the hen houses efficiently. The concept of SOM, which is a data-clustering method to find groups of pullet houses for the transportation management system, is introduced. Once the pullet houses are clustered in the first phase, the differential evolution technique (DE) developed in the DE_G-Q-DVRP-FD algorithm was then used in the second phase for determining routes for transferring pullets from the pullet houses to the hen houses. The performance of the proposed algorithms was measured using the relative improvement (*RI*), which compares the total cost. The results obtained from this study show that the MESOMDE_G-Q-DVRP-FD algorithm provided better total cost values than the firm's current practice by 7.59–31.28 % (with an average of 19.17 %), and 0.84–13.15 % (with an average of 8.07 %) better than the DE_G-Q-DVRP-FD algorithm. Paired sample tests were also performed to compare the cost per unit transported between the two sizes of vehicles of the proposed algorithms and the current practice. Results of the tests showed that the 10-wheel vehicle yields the lower cost per unit transported for both proposed algorithms, and the MESOMDE_G-Q-DVRP-FD algorithm gives the lowest cost per unit transported. Additionally, the MESOMDE_G-Q-DVRP-FD was compared with the benchmark problem named Growing Neural Gas (GNG). The experimental results reveal that the MESOMDE_G-Q-DVRP-FD algorithm is more efficient compared to the GNG algorithm. Therefore, the MESOMDE_G-Q-DVRP-FD algorithm can be used efficiently to solve the problem of pullet transportation to hen houses. This will help the egg industry reduce the total cost resulting in sustainable production.

The proposed MESOMDE_G-Q-DVRP-FD method is not only useful for reducing the total cost when compared to the current practice, but also for efficient management of a poultry production system. Furthermore, the method of this research should prove beneficial to other similar agro-food sectors and other sectors concerned in transportation of commodities in Thailand and around the world. Therefore, there is still a great opportunity to extend our work in many areas. Our planned future work will be to improve the routes from pullet houses to hen houses in order to minimize the transportation cost with consideration of fleet sizes. Another valuable avenue for future research is to consider some other parameters of the problem as fuzzy variables, such as egg demand, fuel cost, speed of vehicles due to road conditions and fleet sizes. We believe that this can add to the ability of our system to model real world problems and will be a valuable extension. Additionally, although the MESOMDE_G-Q-DVRP-FD algorithm has shown an outstanding ability to solve the problem at hand, there is a possibility for other researchers to use hybrid methods or other meta-heuristics to solve the same problem or to design an exact method to compare the strength of various approaches in solving the problem of this nature. Furthermore, the consideration of multiple objective functions may be interesting.

Acknowledgments This work was supported by the Higher Education Research Promotion and National Research University Project of Thailand, Office of the Higher Education Commission, through the Food and Functional Food Research Cluster of Khon Kaen University and in collaboration with the Research Unit on System Modeling for Industry, Khon Kaen University, Thailand.

Appendix

```
begin
Locations of hen houses are represented with 2-D vector $x_i$ from mapping
Create two prototype nodes $w_1$ and $w_2$ with random initialization
for $l = 2 : nw_t$
    for $k = 1 : \lambda_{max}$
    Randomly select an input vector $x_i$
    Find nodes $w_a$ and $w_b$ corresponding to $\| x_i - w_a \|^2 \leq \| x_i - w_b \|^2 \leq \ldots \leq \| x_i - w_l \|^2$
    Update $error_a \leftarrow error_a + \| x_i - w_a \|^2$
    Move node $w_a$ and its topological neighbors (nodes connected to $w_a$) toward $x_i$ :
            $w_a \leftarrow w_a + \varepsilon_w (x_i - w_a)$
            $w_n \leftarrow w_n + \varepsilon_n (x_i - w_n) \quad \forall n \in neighbors(w_a)$
    if Edge between $w_a$ and $w_b$ ( $edge_{a,b}$ ) does not exist then
        create edge between $w_a$ and $w_b$ ( $edge_{a,b}$ )
    endif
    $age_{a,b} \leftarrow 0$ and $age_{(a,n)} \leftarrow age_{(a,n)} + 1 \; \forall n \in neighbors(w_a)$
    Remove all edges with an age larger than $age_{\max}$ and nodes with no edges
    endfor
  Determine the node c with the largest error and the neighbor node d of c with the largest error
  Insert the new node e between c and d: $w_e \leftarrow (w_c + w_d)/2$
  Update the number of nodes: $l \leftarrow l + 1$
  Create edges: $edge_{c,e}$ , $edge_{d,e}$ and remove $edge_{c,d}$
  $error_c \leftarrow \alpha \times error_c$ , $error_d \leftarrow \alpha \times error_d$ and $error_e \leftarrow error_c$
  Decrease all errors of all nodes l: $error_l \leftarrow \beta \times error_l$
  endfor
end;
```

References

Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, *36*(5), 1693–1702.

Anily, S., & Bramel, J. (1999). Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. *Naval Research Logistics*, *46*, 654–670.

Anily, S., & Hassin, R. (1992a). The swapping problem. *Networks*, *22*(4), 419–433.

Archetti, C., Speranza, M. G., & Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, *40*(1), 64–73.

Arnonkijpanich, B., Chaikanha, N., Pathumnakul, S., & Lursinsap, C. (2004). Proportional self-organizing map (PSOM) based on flexible capacity buffer for allocating sugar cane loading stations. In *Proceedings of the IEEE international conference on systems, manufacturing and cybernetics*, pp. 6206–6211.

Arnonkijpanich, B., Hasenfuss, A., & Hammer, B. (2011a). Local matrix adaptation in topographic neural maps. *Neurocomputing*, *74*(4), 522–539.

Belmecheri, F., Prins, C., Yalaoui, F., & Amodeo, L. (2013). Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *Journal of Intelligent Manufacturing*, *24*(4), 775–789.

Berbeglia, G., Cordeau, J. F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *Top*, *15*(1), 1–31.

Boonmee, A., Sethanan, K., & Arnonkijpanich, B. (2013). Growing neural gas approach for minimizing the total cost of hen allocation to poultry farms in Thailand. In *Proceedings of the Asia Pacific industrial engineering and management system*.

Boonmee, A., Sethanan, K., Arnonkijpanich, B., & Theerakulpisut, S. (2015). Minimizing the total cost of hen allocation to poultry farms using hybrid growing neural gas approach. *Computers and Electronics in Agriculture*, *110*, 27–35.

Chakaravarthy, G. V., Marimuthu, S., & Sait, A. N. (2013). Performance evaluation of proposed differential evolution and particle swarm pptimization algorithms for scheduling m-machine flow shops with lot streaming. *Journal of Intelligent Manufacturing*, *24*, 175–191.

Chen, Q., Li, K., & Liu, Z. (2014). Model and algorithm for an unpaired pickup and delivery vehicle routing problem with split loads. *Transportation Research Part E*, *69*, 218–235.

Cordeau, J. F., & Laporte, G. (2003). The dial-a-ride problem (DARP): Variants modeling issues and algorithms. *Journal of the Belgian, French and Italian Operations Research Societies*, *1*, 89–101.

Dechampai, D., Sethanan, K., Tanwanichkul, L., & Arnonkijpanich, B. (2013). Transportation management systems of pullet farm using multifactor based evolving self-organizing maps (MESOM). In *Proceedings of the Asia Pacific industrial engineering and management system*.

Deng, D., & Kasabov, N. (2000). ESOM: An algorithm to evolve self-organizing maps from on-line data streams. *IEEE International Joint Conference on Neural Networks*, *2000*, 3–8.

Dong, G., Tang, J., Lai, K. K., & Kong, Y. (2011). An exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales companies based on set-partitioning model. *Journal of Intelligent Manufacturing*, *22*, 789–799.

Dror, M., & Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, *23*(2), 141–145.

Dror, M., Laporte, G., & Trudeaub, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics*, *50*(3), 239–254.

Erbao, C., & Mingyong, L. (2009). A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands. *Journal of Computational and Applied Mathematics*, *231*, 302–310.

Erbao, C., Mingyong, L., & Kai, N. (2008). A differential evolution & genetic algorithm for vehicle routing problem with simultaneous delivery and pick-up and time windows. In *Proceedings of the international federation of automatic control 17th world congress*, pp. 6–11.

Fritzke, B. (1995). A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, *7*, 625–632.

Hernandez-Perez, H., & Salazar-Gonzalez, J. J. (2004). A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics*, *145*, 126–139.

Hou, L., Zhou, H., & Zhao, J. (2010). A novel discrete differential evolution algorithm for stochastic VRPSPD. *Journal of Computational Information Systems*, *6*(8), 2483–2491.

Karen, I., Kaya, N., & Öztürk., F. (2013). Intelligent die design optimization using enhanced differential evolution and response surface methodology. *Journal of Intelligent Manufacturing*. doi:10.1007/s1084501307951.

Kim, B. I., & Son, S. J. (2012). A probability matrix based particle swarm optimization for the capacitated vehicle routing problem. *Journal of Intelligent Manufacturing*, *23*(4), 1119–1126.

Kohonen, T. (1995). *Self-organizing maps*. New York: Springer.

Lai, M. Y., & Cao, E. B. (2010). An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence*, *23*(2), 188–195.

Landrieu, A., Mati, Y., & Binder, Z. (2001). A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing*, *12*, 497–508.

Ma, W., Wang, M., & Zhu, X. (2013). Hybrid particle swarm optimization and differential evolution algorithm for bi-level programming problem and its application to pricing and lot-sizing decisions. *Journal of Intelligent Manufacturing*,. doi:10.1007/s1084501308035.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, pp. 281–297.

Martinovic, G., Aleksi, I., & Baumgartner, A. (2008). Single-commodity vehicle routing problem with pickup and delivery service. *Mathematical Problems in Engineering*. doi:10.1155/2008697981.

Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, *162*(1), 126–141.

Nearchou, A. C. (2006). Meta-heuristics from nature for the loop layout design problem. *International Journal of Production Economics*, *101*(2), 312–328.

Şahin, M., Çavuşlar, G., Öncan, T., Şahin, G., & Aksu, D. T. (2013). An efficient heuristic for the multi-vehicle one-to-one pickup and delivery problem with split loads. *Transportation Research Part C*, *27*, 169–188.

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*, 341–359.

Vahdani, B., Tavakkoli-Moghaddam, R., Zandieh, M., & Razmi, J. (2012). Vehicle routing scheduling using an enhanced hybrid optimization approach. *Journal of Intelligent Manufacturing*, *23*(3), 759–774.

Vincent, L. W. H., Ponnambalam, S. G., & Kanagaraj, G. (2012). Differential evolution variants to schedule flexible assembly lines. *Journal of Intelligent Manufacturing*. doi:10.1007/s1084501207168.

Wang, C. H., & Lu, J. Z. (2010). An effective evolutionary algorithm for the practical capacitated vehicle routing problems. *Journal of Intelligent Manufacturing*, *21*(4), 363–375.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. *Expert Systems with Applications*, *36*(2), 1070–1081.