EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

# Differential evolution algorithms with local search for the multi-products capacitated vehicle routing problem with time windows: A case study of the ice industry

Rattanapong Kamphukaew[1)], Kanchana Sethanan*[1)], Thitipong Jamrus[1)] and Hung-kai Wang[2)]

[1)]Research Unit on System Modeling for Industry, Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand
[2)]Department of Industrial Engineering and Systems Management, Feng Chia University, Taichung 40724, Taiwan

Received 6 June 2018
Accepted 28 August 2018

**Abstract**

This research aims to offer a solution to the capacitated vehicle routing problem for multiple products, time windows and a heterogeneous fleet for ice transport. The problem structure is that the capacitated vehicle routing problem is one-to-many with multiple products, where each customer has a variety of product types and demands. Also, limited delivery time is considered for each customer. We used a mixed integer linear programming model to give an optimal solution, and propose a differential evolution method with a local search that we have developed. The objective is to sequence the delivery procedures for ice transport to minimize total costs, comprised of traveling costs, driver wages and a penalty costs. We obtained various solutions by a comparison of our proposed solution and the optimal solution of the ice transport case study and differential evolution with local search (DELS) to validate the proposed metaheuristic. The relative improvement between the current practice of the ice transport in the case study and the differential evolution with a local search was 3.70-25.75%. The differential evolution with a local search outperformed current practices by an average by 2.29%.

**Keywords:** Ice transportation, Capacitated vehicle routing problem, Time windows, Differential evolution algorithm, Local search

## 1. Introduction

Ice is an essential product for the preservation of Thailand's agricultural products because of its tropical and very humid climate. Customers require several types of ice products. The ice industry has very high costs in two particular areas, production costs and transportation systems. In terms of production, Thailand's ice industry has a mostly inefficient production management system, resulting in a high proportion of waste. Another major problem of the ice industry is transportation management. The ice industry needs significantly reduced transportation and delivery costs. Inefficient transportation management can cause high transport costs and ice melting during shipment. Another major cost incurred is the cost of transportation itself. Delivery in this ice industry case study comes directly from a factory. Its customers are restaurants, cafés and general stores.

The transportation problems of the ice industry are an important part of the supply chain. In the transportation process, temperature controlled vehicles are most often used, having refrigeration equipment installed in the cargo area located at the back of the vehicle. These vehicles are usually expensive and consume more fuel than more modern vehicles. Ice usually has a short shelf life, so it is necessary to use temperature-controlled vehicles to prevent melting during shipment to customers. The route of temperature-controlled vehicles enables delivery of ice directly from the factory to the end-user. The product types consist of tube ice, small tube ice, flake ice, and cube ice. Also, service of the demand has a time limit, and the driver has to serve the customers in a limited time interval. This problem is not well managed, in terms of routing and efficient delivery systems. The ice industry needs improvements and reduction of transportation costs, as well as greatly improved delivery systems deliver in the time window of customers.

Thus, this research proposes two solutions to solve the problem. (First, a mathematical model of multi-product transport, vehicle capacity and time window is proposed as the best solution for small problems. If the problem is larger, there are more variables. Solving the problem with mathematical models may require more computational resources. Therefore, a metaheuristic algorithm was developed to solve these problem as a second method. Second, a differential evolution algorithm was used to address the issue of routing ice delivery using temperature-controlled vehicles. The solution was improved using a differential evolution algorithm with a local search in multi-product capacitated vehicle routing with time window for the ice delivery. Temperature-controlled vehicle routes were

*Corresponding author. Tel: +6681 553 6429
Email address: skanch@kku.ac.th
doi: 10.14456/easr.2018.37

considered for delivery to the customer with lowest total costs for ice delivery, considering traveling, driver wages and penalty costs.

The vehicle routing problem has been extended in many ways by introducing additional real-life transportation situations, resulting in many variants of the vehicle routing problem [1]. It is possible to extend this problem by varying vehicle capacities, which results in a heterogeneous fleet vehicle routing problem. The vehicle routing problem is a well-known problem in operations research, logistics and transportation optimization. Dantzig and Ramser (1959) [2] are credited with being the first to introduce the "truck dispatching problem". Many researchers have considered the vehicle routing problem in various industries, for example Worasan et al. (2014) [3] developed a mixed integer programming model to solve a raw milk collection problem. Their objective was to minimize the total costs, which were comprised of transportation and cleaning costs. Akpinar (2016) [4] proposed a hybrid algorithm that executes a large neighborhood search in combination with the ant colony optimization algorithm for the capacitated vehicle routing problem.

The capacitated vehicle routing problem for multiple products is an extension of the classic capacitated vehicle routing problem, where various products are transported together in one vehicle. This is encountered in many industries, such as delivery of food and groceries, garbage collection, and marine products [5]. Several researchers have considered this problem. For example, Zhang and Chen (2014) [6] presented the capacitated vehicle routing problem for multiple products encountered in the frozen food delivery industry. They managed the delivery of a variety of products. The objective was to find the routes that represent the minimum delivery cost for a fleet of identical vehicles. The problem was solved using a genetic algorithm. A vehicle routing problem with a time window (VRPTW) is a CVRP, but the VRPTW adds consideration of delivery time. The VRPTW is not only a combinatorial problem, but it is also more complex when it considers time windows. There is research about VRPs with time windows. For example, Hsu et al. ( 2007) [7] presented a VRPTW considering the randomness of the perishable food delivery process and they proposed VRPTW models. The results indicate that inventory and energy costs can significantly influence total delivery costs. Also, Osvald and Stirn (2008) [8] considered a vehicle routing problem with time windows and time-dependent travel times. They considered the impact of perishability as part of the overall distribution costs and a heuristic approach, based on a tabu search, used to solve the problem. Jiang et al. (2014) [9] proposed a two-phase solution method that extends an existing tabu search procedure to solve a VRPTW. However, it was observed that the search process would sometimes not be able to avoid a local optimum.

Recently, metaheuristics such as genetic algorithms, particle swarm optimization and differential evolution have been successfully applied in several fields. For example, production scheduling (Chamnanlor and Sethanan 2015 [10], Jamrus et al. 2018 [11]), the generalized assignment problem (Sethanan and Pitakaso 2016 [12]) and scheduling vehicle problems (Zhang and Chen 2014 [6]).

Differential evolution is a population based search technique, which uses simple operations on a generated set of initial vectors, mutation, crossover and selection, to create new competitive solutions. Its straightforward nature, quick convergence, and other features make it very attractive for numerical optimization. The key difference between differential evolution and genetic algorithms or particle swarm optimization is in a unique mechanism for generating new solutions that are generated by combining several other solutions with the candidate solution [13]. Differential evolution is one of the most powerful techniques that has been effectively applied to continuous optimization, such as machine layout, manufacturing problems, warehouse storage location assignment problems and scheduling. For example, Nearchou (2006) [14] proposed using a differential evolution algorithm to design machine layouts, the unidirectional loop layout design problem, in a flexible manufacturing system. Wisittipanich and Hengmeechai (2015) [15] presented a multi-objective differential evolution for door assignment and truck scheduling in a multiple inbound and outbound door cross-docking problem according to the Just-In-Time concept.

Even though differential evolution has been used effectively in a variety of fields, it has been very limited when applied to solving vehicle routing problems, especially the capacitated vehicle routing problem. This is because the encoding pattern of differential evolution cannot be directly applied to the capacitated vehicle routing problem. Xu and Wen (2012) [16] proposed a differential evolution algorithm to solve the vehicle routing problem, considering vehicle capacity restrictions, the longest distance restriction, and fully loaded vehicles. The objective was to find a set of routes to minimize the non-fully loaded factor and the total traveling distance of all vehicles. Later, Lingjuan and Zhijiang (2013) [17] presented a novel discrete differential evolution algorithm for the vehicle routing problem with simultaneous pickups and deliveries. The objective was to find route optimization using their algorithm. Their result was better than the traditional differential evolution algorithm and the existing genetic algorithm. Recently, Dechampai et al. (2017) [18] proposed a capacitated vehicle routing problem with the flexibility of mixing pickup and delivery services, and the maximum duration of a route in the poultry industry. This was done to determine the routing for transferring pullets from pullet houses to hen houses using a differential evolution algorithm. Alinaghian et al. (2015) [19] proposed the green Clark and Wright algorithm and the differential evolution algorithm to solve the green vehicle routing problem capacity restriction. The objective function minimizes total costs, consisting of reduction of fuel costs, driver costs, and the costs of vehicle usage. Moreover, Ponnambalam and Ganesan (2015) [20] developed a differential evolution algorithm in combination with a local search to solve the capacitated vehicle routing problem, minimizing total traveling costs. The results of the differential evolution with local search algorithm gave quality solutions for the capacitated vehicle routing problem.

There are published reports that describe solutions to the capacitated vehicle routing problem and capacitated vehicle routing problem with multiple products. These are summarized in Table 1. The complexity of the capacitated vehicle routing problem with multiple products needs an efficient solution for minimizing total costs. Thus, this paper proposes a differential evolution with a local search for solving the ice transportation problem.

**Table 1** A comparison of the literature on restrictions and approaches

| Researchers | Restrictions | | | Approaches | | | |
|---|---|---|---|---|---|---|---|
| | **Capacity** | **Multi - product** | **Time window** | **Mathematical** | **Heuristic** | **Metaheuristic** | **Local search** |
| Worasan *et al.* (2014) | ✓ | | | ✓ | | ✓ | |
| Akpinar *et al.* (2016) | ✓ | | | ✓ | ✓ | ✓ | |
| De *et al.* (2013) | ✓ | ✓ | | ✓ | | ✓ | |
| Zhang and Chen (2014) | ✓ | ✓ | | ✓ | | ✓ | |
| Hsu et al. (2007) | ✓ | | ✓ | ✓ | ✓ | | |
| Osvald and Stirn (2008) | ✓ | | ✓ | ✓ | | | ✓ |
| Jiang et al. (2014) | ✓ | | ✓ | | ✓ | | |
| Meesuptaweekoon and Chaovalitwongse (2014) | ✓ | | | ✓ | ✓ | | |
| Joshi *et al.* (2015) | ✓ | | | ✓ | ✓ | | |
| Xu and Wen (2012) | ✓ | | | ✓ | | ✓ | |
| Lingjuan and Zhijiang (2013) | ✓ | | | ✓ | | ✓ | |
| Dechampai *et al.* (2017) | ✓ | | | ✓ | ✓ | ✓ | |
| Alinaghian *et al*. (2015) | ✓ | | | ✓ | ✓ | ✓ | |
| Ponnambalam and Ganesan (2015) | ✓ | | | ✓ | | ✓ | ✓ |
| Our study | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |

## 2. The vehicle routing model for the ice industry

### *2.1 Assumptions and constraints*

Based on the above analysis, the premise conditions of the transportation vehicle routing model are as follows.

(1) The logistic depot is known, and the customer points are also known.

(2) The sum of customer demands should not exceed the maximum capacity of the transport vehicle.

(3) Each vehicle takes the depot as the starting point, travels along the distribution route to deliver the ice to the customer-designated locations, and then returns to the depot.

(4) Every customer is serviced by one vehicle, but one vehicle can serve multiple customers and there are no partial shipments.

(5) The transportation vehicle should serve according to the customer time interval, that is, delivery within the time specified by the customers, otherwise there will be a penalty.

(6) The distribution path of vehicles should minimize the total costs as far as possible, while meeting the customer requirements.

### *2.2 Mathematical model*

A mathematical model was developed such that the total costs, i.e., transportation, driver wages and penalty costs of ice products of a vehicle were minimized. Parameters and decision variables used in formulating the model were defined. A 0–1 mixed integer programming formulation is presented below with a brief explanation of each constraint.

*Indices*

| | |
|---|---|
| $i, j, a$ | Customer; $i, j, a = 1, 2, ...,$N |
| $k$ | Temperature-controlled vehicles; $k = 1, 2, ...,$K |
| $p$ | Ice compartment index; $p = 1, 2, ...,$P |

*Input parameters*

| | |
|---|---|
| N | Number of customers |
| K | Number of temperature-controlled vehicles |
| P | Number of products |
| $q_{ip}$ | Amount of product $p$ at customer $i$ (units) |
| $tq_i$ | Total amount for customer $i$ (units) |
| $d_{ij}$ | Traveling distance from customer $i$ to $j$ (km) |
| $t_{ij}$ | Traveling time from customer $i$ to $j$ (hours) |
| $e_i$ | The start time of the period of customer $i$ |
| $l_i$ | The end time of the period of customer $i$ |
| $s_i$ | Service time at customer $i$ (hours) |
| $c_k$ | Capacity of vehicle $k$ (units) |
| $r_k$ | Fuel cost of vehicle $k$ (baht/km) |
| $oc_k$ | Driver wage cost of vehicle $k$ (baht) |
| $pc$ | Penalty cost of customers (baht/unit) |

*Decision variables*

| | |
|---|---|
| $w_i$ | = 1; Missed delivery at customer $i$<br>= 0; Otherwise |
| $x_{ijk}$ | = 1; Vehicle $k$ travels from customer $i$ to customer $j$<br>= 0; Otherwise |
| $y_k$ | = 1; Vehicle $k$ active<br>= 0; Otherwise |
| $z_{ikp}$ | = 1; Vehicle $k$ delivers product $p$ to customer $i$<br>= 0; Otherwise |
| $t_{ikp}$ | Amount of product $p$ shipped by vehicle $k$ at customer $i$ (units) |
| $ta_j$ | Time of vehicle arriving at customer $j$ (hours) |
| $s_{ikp}$ | Decision variables for sub-tour |

*Objective function*

Minimize $Z$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K} d_{ij} r_k x_{ijk} + \sum_{k=1}^{K} oc_k\, y_k + \sum_{i=1}^{N} pc\, tq_i w_i \tag{1}$$

*Constraints*

$$\sum_{j=1}^{N} x_{1jk} \leq 1 \qquad \forall k \tag{2}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ijk} \leq \mathrm{N}\, Y_k \qquad \forall k \tag{3}$$

$$\sum_{i=1}^{N} x_{iak} - \sum_{i=1}^{N} x_{ajk} = 0 \qquad \forall a, k \tag{4}$$

$$\sum_{k=1}^{K} z_{ikp} = 1 \qquad \forall i,p \tag{5}$$

$$\sum_{i=1}^{N}\sum_{p=1}^{P} z_{ikp} q_{ip} \leq c_k \qquad \forall k \tag{6}$$

$$\sum_{j=1}^{N} x_{jik} \geq z_{ikp} \qquad \forall i,k,p \tag{7}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{N} x_{ijk} \leq 1 \qquad \forall j \tag{8}$$

$$\sum_{i=1}^{N} z_{ikp} q_{ip} \leq t_{ikp} \qquad \forall k,p \tag{9}$$

$$\sum_{k=1}^{K} x_{ijk} = w_i \qquad \forall i,j \ when\ i=j \tag{10}$$

$$ta_j \geq (ta_i + s_i + t_{ij}) * x_{ijk} - (l_i * (1 - x_{ijk})) \ \forall i,j,k \tag{11}$$

$$e_j \leq ta_{jk} \leq l_j \qquad \forall j,k \tag{12}$$

$$(ta_j + s_j + t_{1j})\, x_{ijk} \leq Tmax \qquad \forall j,k \tag{13}$$

$$S_{kip} \geq S_{kjp} + q_{ip} - c_k + \left(c_k * \left(x_{ijk} + x_{jik}\right)\right) - x_{ijk}\left(q_{ip} + q_{jp}\right) \quad \forall i,j,p,k \tag{14}$$

$$S_{ikp} \leq c_k - x_{i0k}\left(c_k - q_{ip}\right) \qquad \forall i,p,k \tag{15}$$

$$S_{ikp} \leq q_{ip} + \sum_{i=1}^{N} x_{jik} q_{ip} \qquad \forall k,p \tag{16}$$

The objective function (1) minimizes the total costs. It consists of distance travelled by the vehicles, driver wages and penalty costs. In this research, the penalty cost adopts a time window constraint, which allows for arrival at a time outside the window with a penalty. We assumed that the penalty cost is correlated to the price of ice and the lateness of the deliver. Constraint (2) ensures that vehicle k starts its travel from the depot to customer *i*. Constraint (3) ensures there are *k* vehicles active for delivery to customers. Constraint (4) ensures that if a vehicle delivers to a customer, that vehicle must leave from that customer. Constraint (5) ensures that a vehicle must visit customers, if those customers have demand for products to be delivered. Constraint (6) ensures that the total demand for product p by customer *i* uploaded by vehicle *k* must not exceed the capacity of that vehicle k. Constraint (7) ensures that vehicle *k* is able to travel to customer *i* with product *p* when vehicle *k* travels from customer *i*. Constraint (8) guarantees that any customer *j* was serviced by vehicle *k*. Constraint (9) describes the total demand for customer *i* all uploaded by vehicle *k* must not exceed the amount of product *p* shipped. Constraint (10) is a time outside window condition. Constraint (11) ensures any vehicle *k* delivers within the time window of the customers. Constraint (12) ensures vehicle *k* returns to the depot within the time window. Constraint (13) ensures a penalty when vehicle *k* delivers outside the time window of customer *i*. Constraints (14), (15) and (16) ensure sub-tour conditions.

## 3. Differential evolution with local search (DELS)

### *3.1 Differential Evolution Algorithm (DE)*

Price and Strom (1997) first introduced the differential evolution algorithm [21]. This algorithm has four methods consisting of generation of an initial solution, mutation, recombination, and selection. For this case study, we used a metaheuristic that finds optimal solutions and applied them to complex and large problems. Therefore, we solved problems by a metaheuristic approach based on the original differential evolution. The differential evolution algorithm method is given in the form of an algorithm as shown in Figure 1.

**Procedure:** Differential evolution algorithm with local search
**Input:** capacitated vehicle routing problem multiple products problem data, differential evolution parameters (CR, F, NP)
**Output:** Optimal solution
**Begin:**
    Encoding step: randomly generate a set of target vectors *n* (*n* =1…*NP*); // *n* is the index of the vector representation; *NP* is a predefined number of the population
    **while** termination condition is not satisfied **do**
        **for** *n* = 1 to *NP* //*NP* is the predefined number of population
            Solution Initialization
            Decoding and Evaluation
            Mutation Operation
            Recombination Operation
            Selection Operation
        **end**
            Insert neighborhood (local search)
    **end**
**end;**

**Figure 1** Overall differential evolution algorithm with local search procedure for the capacitated vehicle routing problem with multiple products and time window

The number of the samples in the population is the population size or the number of vectors that will be used in differential evolution iteration. The number of samples in the population will be randomly generated in the first iteration and each vector is similar. The vectors will all be transformed into mutant and trial vectors using mutation and recombination processes. The vectors consist of the delivery sequence of customers and types of vehicle for delivery as shown in Figure 2.

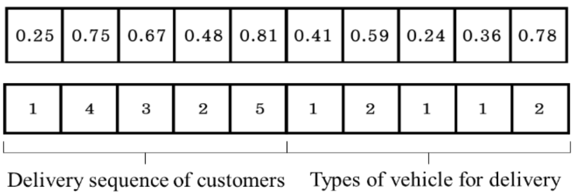


**Figure 2** Example of randomly generated encoding and decoding.

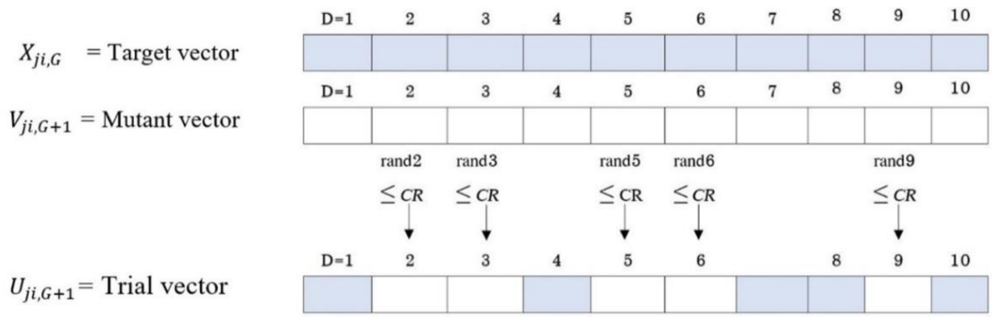


**Figure 3** Example of recombination

The mutation operation is the second process of the differential evolution mechanism. Equation (17) was used to randomly combine three selected vectors into a mutant vector. The scaling factor is a constant from [0, 2]. $V_{i,G}$ is called the mutant vector.

$$V_{ji,G+1} = X_{r1,G} + F\,(X_{r2,G} - X_{r3,G}) \tag{17}$$

where

$V_{ji,G+1}$ = Mutant vector
$X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$ = Random vector
$F$ = Scaling factor

Recombination incorporates successful solutions from the previous iteration as shown in Figure 3. A trial vector $U_{ji,G+1}$ is obtained from Equation (18). The position value of a vector can be the position value of a target or trial vector, depending on a random number that is generated for that position, and the value of the crossover rate. The value at a specified position will be copied from the value of that position in the target vector, if the random number for that position is greater than the crossover rate. Alternatively, if the random number generated for that position is lower than the crossover rate, the position value at that position will be equal to the value of the mutant vector. The formula that represents this mechanism is Equation (18).

$$U_{ji,G+1} = \begin{cases} V_{ji,G+1} & \text{if } (\text{randb}(j) \leq CR) \text{ or } j = \text{rnbr}(i) \\ X_{ji,G} & \text{if } (\text{randb}(j) > CR) \text{ or } j \neq \text{rnbr}(i) \end{cases} \tag{18}$$

where

$U_{ji,G+1}$ = Trial vector
$V_{ji,G+1}$ = Mutant vector
$X_{ji,G}$ = Target vector
$\text{randb}(j)$ = A random number generated from [0, 1]
$\text{rnbr}(i)$ = A random integer from [1, 2,…, D]
$CR$ = The crossover rate [0, 1]

*3.2 Improved solution by local search (LS)*

Local search is a heuristic method for solving computationally hard optimization problems. This research used the insert neighborhood method for improvement of the DELS solution. The initial DELS solution used the vector from a processed DE that inserted positions by distance movement. This method was applied from the flexible flow shop scheduling problem by Laguna et al. (1991) [22] and Laguna et al. (1993) [23]. Equations (19) and (20) were used to find the distance of movement for a number of customers less than or equal to 30 and greater than 30 customers, respectively. When we insert the neighborhood process into the vectors, we evaluate the vectors again as for the initial solution. The best vectors were selected as an initial solution for the next iteration.

$$dm = [n/2]-1 \tag{19}$$

$$dm = ([n/2]/2)*c/4 \tag{20}$$

where

n = Number of customers
c = Constant number between [1,4] by Laguna et al. (1991)

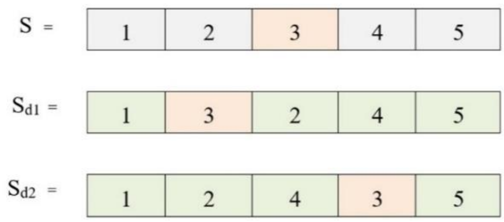


**Figure 4** Example of a local search

After the neighborhood condition is inserted for distance movement, the solution is evaluated in the same way as the previous local search process as shown in Figure 4. The DELS provides various solutions and is able to improve solutions that it keeps as a best solution. Then, the next iteration will be selected for the best solution by comparison between the DELS vectors and target vectors.

## 4. Experiment and results

This section describes experiments using the mathematical model, differential evolution algorithms and differential evolution with a local search to demonstrate efficiency and effectiveness. For illustration, we generated 27 problems (Table 2) involving various numbers of customers, demands and time windows. The vehicle types are large and small in this company, and product types are tube ice and flake ice. This experiment generated 27 problems for presentation with five iterations as the route sequences and the objective function results of each problem. Optimal solutions were found using the LINGO optimization program, while differential evolution and differential evolution with local search were developed using MATLAB (R2017a) on a 2.10 GHz PC, with 8 G-Byte of RAM, for testing and evaluation.

The experiment for this research was designed for testing three factors, the numbers of customers, demands and time windows. The combination experiments were conducted in five replicates. The parameters were divided into three levels: low (L), medium (M) and high (H) and considered to include the number of customers (10, 50 and 100 customers),

**Table 2** The experiment of problems with various levels

| Problem | Replication | Factors | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Customers | | | % Uncertain demands | | | Time window | | |
| | | L 10 | M 50 | H 100 | L 10 | M 30 | H 50 | L 1 | M 3 | H 5 |
| 1 | 5 | ✓ | | | ✓ | | | ✓ | | |
| 2 | 5 | ✓ | | | ✓ | | | | ✓ | |
| 3 | 5 | ✓ | | | ✓ | | | | | ✓ |
| 4 | 5 | ✓ | | | | ✓ | | ✓ | | |
| 5 | 5 | ✓ | | | | ✓ | | | ✓ | |
| 6 | 5 | ✓ | | | | ✓ | | | | ✓ |
| 7 | 5 | ✓ | | | | | ✓ | ✓ | | |
| 8 | 5 | ✓ | | | | | ✓ | | ✓ | |
| 9 | 5 | ✓ | | | | | ✓ | | | ✓ |
| 10 | 5 | | ✓ | | ✓ | | | ✓ | | |
| 11 | 5 | | ✓ | | ✓ | | | | ✓ | |
| 12 | 5 | | ✓ | | ✓ | | | | | ✓ |
| 13 | 5 | | ✓ | | | ✓ | | ✓ | | |
| 14 | 5 | | ✓ | | | ✓ | | | ✓ | |
| 15 | 5 | | ✓ | | | ✓ | | | | ✓ |
| 16 | 5 | | ✓ | | | | ✓ | ✓ | | |
| 17 | 5 | | ✓ | | | | ✓ | | ✓ | |
| 18 | 5 | | ✓ | | | | ✓ | | | ✓ |
| 19 | 5 | | | ✓ | ✓ | | | ✓ | | |
| 20 | 5 | | | ✓ | ✓ | | | | ✓ | |
| 21 | 5 | | | ✓ | ✓ | | | | | ✓ |
| 22 | 5 | | | ✓ | | ✓ | | ✓ | | |
| 23 | 5 | | | ✓ | | ✓ | | | ✓ | |
| 24 | 5 | | | ✓ | | ✓ | | | | ✓ |
| 25 | 5 | | | ✓ | | | ✓ | ✓ | | |
| 26 | 5 | | | ✓ | | | ✓ | | ✓ | |
| 27 | 5 | | | ✓ | | | ✓ | | | ✓ |

**Table 3** The results from design of experiments using an optimal solution and DELS

| Problem | No. customers | % Uncertain demands | Time window [hours] | Total cost [Baht] | | CPU time [sec] | | *DELS_EF* (%) |
|---|---|---|---|---|---|---|---|---|
| | | | | Optimal | DELS | Optimal | DELS | |
| 1 | 10 | 10 | 1 | 1,483.10 | 1,539.90 | 10 | 38.17 | 96.31 |
| 2 | 10 | 10 | 3 | 1,400.10 | 1,485.90 | 65 | 39.86 | 94.23 |
| 3 | 10 | 10 | 5 | 1,331.60 | 1,345.50 | 2,914 | 34.38 | 98.97 |
| 4 | 10 | 30 | 1 | 1,680.40 | 1,707.50 | 144 | 36.52 | 98.41 |
| 5 | 10 | 30 | 3 | 1,547.80 | 1,556.70 | 981 | 39.62 | 99.43 |
| 6 | 10 | 30 | 5 | 1,508.80 | 1,589.80 | 3,377 | 35.72 | 94.91 |
| 7 | 10 | 50 | 1 | 1,859.30 | 1,872.30 | 208 | 38.34 | 99.31 |
| 8 | 10 | 50 | 3 | 1,785.70 | 1,810.20 | 501 | 42.62 | 98.65 |
| 9 | 10 | 50 | 5 | 1,744.90 | 1,777.40 | 13,353 | 35.07 | 98.17 |
| 10 | 50 | 10 | 1 | n/a* | 12,916.30 | - | 196.44 | - |
| 11 | 50 | 10 | 3 | n/a* | 8,606.50 | - | 196.51 | - |
| 12 | 50 | 10 | 5 | n/a* | 7,771.30 | - | 306.35 | - |
| 13 | 50 | 30 | 1 | n/a* | 11,883.10 | - | 205.52 | - |
| 14 | 50 | 30 | 3 | n/a* | 9,459.10 | - | 209.59 | - |
| 15 | 50 | 30 | 5 | n/a* | 8,549.50 | - | 297.53 | - |
| 16 | 50 | 50 | 1 | n/a* | 13,258.80 | - | 211.84 | - |
| 17 | 50 | 50 | 3 | n/a* | 9,631.20 | - | 199.75 | - |
| 18 | 50 | 50 | 5 | n/a* | 9,066.40 | - | 314.71 | - |
| 19 | 100 | 10 | 1 | n/a* | 31,870.60 | - | 409.82 | - |
| 20 | 100 | 10 | 3 | n/a* | 21,854.60 | - | 401.52 | - |
| 21 | 100 | 10 | 5 | n/a* | 15,879.50 | - | 468.97 | - |
| 22 | 100 | 30 | 1 | n/a* | 32,631.50 | - | 429.52 | - |
| 23 | 100 | 30 | 3 | n/a* | 20,6290 | - | 444.92 | - |
| 24 | 100 | 30 | 5 | n/a* | 16,109.70 | - | 519.27 | - |
| 25 | 100 | 50 | 1 | n/a* | 29,412.40 | - | 456.42 | - |
| 26 | 100 | 50 | 3 | n/a* | 21,7800 | - | 477.63 | - |
| 27 | 100 | 50 | 5 | n/a* | 18,243.50 | - | 643.25 | - |
| | | | Average | | | | | 97.80 |

fluctuation of demand (10%, 30%, and 50% of ice demands, which was at an average of 30 ice products) and time windows with starting and ending times (1, 3, and 5 hours). In total there were 135 samples and the results are shown the percentage of improvement presented in Equations 21 and 22.

$$DELS_{EF(\%)} = \frac{R_{OPT}}{R_{DELS}} \times 100 \qquad (21)$$

where
$DELS\_EF$ = Efficiency of DELS (%)
$R\_OPT$ = The result of best solution by the mixed integer programming (baht per day)
$R\_DELS$ = The DELS results (baht per day)

$$RI(\%) = \frac{(R_{DE} - R_{DELS})}{R_{DE}} \times 100 \qquad (22)$$

where
$RI$ = Relative improvement between DE and DELS (%)
$R\_DE$ = The DE results (baht per day)
$R\_DELS$ = The DELS results (baht per day)

We formulated the problems using a mixed- integer programming model, differential evolution algorithm and differential evolution with local search based on a practical problem size for the capacitated vehicle routing problem with multiple products, as shown in Tables 3 and 4.

Table 3 shows a comparison of the efficiencies of the best solution using mixed integer programming and the DELS results. For instances 1 to 9, the comparison of average efficiency between the best solution and the DELS results is 97.80%. As seen in the results, an optimal solution cannot be obtained by mixed integer programming in large size problems, due to the vehicle routing problem being NP-hard and highly complex. The DELS gives an acceptable solution compared to the optimal solution. We plotted the evolution of the best solution found by DELS during a typical execution when solving problem 3. As shown in Figure 5, there is a fast run time and convergence rate towards accurate solutions at the beginning of the execution, with the best solution of DELS being 1,345.50 baht and using about 34.38 seconds. The optimal solution is not that fast with the best solution by mixed integer programming being 1,331.60 baht and taking about 2,914 seconds. The compared average efficiency between the best solution and the DELS result of problem 3 is 98.97%.

Table 4 shows that the DELS is better than the DE. The relative improvement in the total cost averaged 3.73%. Differential evolution with a local search shows that small problems have highest relative improvement, while large sized problems show less improvement. However, the differential evolution and the differential evolution with a local search also

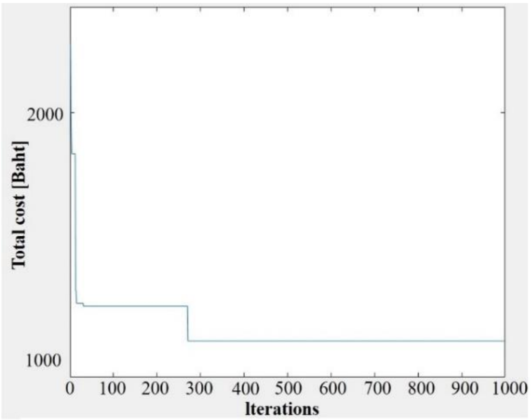


**Figure 5** Typical execution plot for problem 3

**Table 4** The results from design of experiments using DE and DELS

| Problem | No. of customers | % Uncertain demand | Time window [hours] | Total cost [Baht] | | CPU time [sec] | | RI |
|---|---|---|---|---|---|---|---|---|
| | | | | DE | DELS | DE | DELS | (%) |
| 1 | 10 | 10 | 1 | 1590.88 | 1561.66 | 13.30 | 33.49 | 1.84 |
| 2 | 10 | 10 | 3 | 1531.32 | 1518.84 | 14.32 | 37.36 | 0.81 |
| 3 | 10 | 10 | 5 | 1444.82 | 1378.24 | 13.16 | 33.90 | 4.61 |
| 4 | 10 | 30 | 1 | 1,843.76 | 1,757.70 | 14.28 | 38.30 | 4.67 |
| 5 | 10 | 30 | 3 | 1,700.72 | 1,613.92 | 15.09 | 42.39 | 5.10 |
| 6 | 10 | 30 | 5 | 1,644.32 | 1,603.04 | 13.41 | 34.80 | 2.51 |
| 7 | 10 | 50 | 1 | 1,896.90 | 1,890.22 | 14.90 | 38.44 | 0.35 |
| 8 | 10 | 50 | 3 | 1,876.82 | 1,864.70 | 15.32 | 43.63 | 0.65 |
| 9 | 10 | 50 | 5 | 1,784.44 | 1,770.42 | 14.04 | 36.51 | 0.79 |
| 10 | 50 | 10 | 1 | 15,334.28 | 13,474.34 | 39.55 | 206.64 | 12.13 |
| 11 | 50 | 10 | 3 | 9,566.02 | 9,095.72 | 40.19 | 207.40 | 4.92 |
| 12 | 50 | 10 | 5 | 7,994.08 | 7,889.82 | 42.95 | 310.00 | 1.30 |
| 13 | 50 | 30 | 1 | 14,759.54 | 13,781.42 | 40.73 | 216.58 | 6.63 |
| 14 | 50 | 30 | 3 | 9,980.92 | 9,838.16 | 40.01 | 217.03 | 1.43 |
| 15 | 50 | 30 | 5 | 8,815.36 | 8,628.54 | 42.87 | 324.57 | 2.12 |
| 16 | 50 | 50 | 1 | 15,148.92 | 13,819.34 | 40.20 | 218.21 | 8.78 |
| 17 | 50 | 50 | 3 | 10,390.90 | 9,876.90 | 40.26 | 214.52 | 4.95 |
| 18 | 50 | 50 | 5 | 9,402.00 | 9,258.36 | 42.96 | 335.02 | 1.53 |
| 19 | 100 | 10 | 1 | 35,726.32 | 32,320.16 | 84.43 | 426.77 | 9.53 |
| 20 | 100 | 10 | 3 | 23,499.06 | 22,455.32 | 83.82 | 406.49 | 4.44 |
| 21 | 100 | 10 | 5 | 16,097.12 | 15,930.04 | 85.62 | 501.58 | 1.04 |
| 21 | 100 | 30 | 1 | 35,712.18 | 34,245.02 | 83.83 | 449.58 | 4.11 |
| 23 | 100 | 30 | 3 | 23,667.20 | 22,135.82 | 83.92 | 470.03 | 6.47 |
| 24 | 100 | 30 | 5 | 16,981.64 | 16,674.78 | 85.51 | 598.39 | 1.81 |
| 25 | 100 | 50 | 1 | 34,087.58 | 32,802.60 | 84.83 | 461.53 | 3.77 |
| 26 | 100 | 50 | 3 | 24,514.38 | 23,595.98 | 84.37 | 492.05 | 3.75 |
| 27 | 100 | 50 | 5 | 18,640.32 | 18,497.80 | 86.09 | 654.77 | 0.76 |
| | | | | | | | Average | 3.73 |

provided lower a total cost with relatively fast computing times that were less than the current practice of the ice industry case study. The metaheuristic provides various solutions and can improve upon them, keeping the best solution.

## 5. Discussion

In this paper, we present solutions to the multi-product capacitated vehicle routing problem with a time window by a mathematical model, differential evolution algorithm and differential evolution with a local search. The objective is to sequence the delivery of vehicles for the ice industry to minimize total costs, which are comprised of transportation costs, driver wages and penalty costs. In a small-sized problem (instances 1 to 9), for which a solution could be found by LINGO, the average total cost was 1,593.52 baht, while the DELS was able to find optimal solutions equal to those obtained by a LINGO optimization solver. For solving larger-sized problems (instances 10 to 27), a solution could not be solved using LINGO, since the number of variables was excessive due to the vehicle routing problem being NP-hard and highly complex. However, the DELS provided a lower total cost with relatively faster computing times than mixed integer linear programming. The metaheuristic provided various solutions and can improve its solutions, since it keeps the best solution. Additionally, the DE adds a local search that provides more varied solutions to find the best solution. For a particular comparison, the percentages of relative improvement between the solution, the DE and the DELS outperformed the solution in all 27 problems by 0.35-12.13%. We can expect the quality obtained when applying the DE algorithm and the DELS to solve problems with larger numbers of customers or different numbers of test programs. The results show that the proposed DE converged to the global search procedure to find the solution quickly and the optimal solution is acceptable, due to the floating-point numbers used in the calculation.

However, one limitation of this study is that the demand of each customer for each product is constant and known in advance. Vehicles are partitioned into a fixed number of compartments with certain vehicle capacities. Customers are assigned to routes so that the total demand of customers delivered along any route for a certain product does not exceed the capacity of the compartment reserved for ice products. Therefore, future study can extend the total demand of customers receiving deliveries from more than one vehicle (split demand) and the proposed modifications of the DELS, and use other metaheuristics or hybrid methods to improve the solutions. Also, more studies can be done to develop approaches for the capacitated vehicle routing problem for multiple products with periodicities, and consider other ice distribution costs such as freezing cost and ice damage cost for a real-life ice distribution system.

## 6. Conclusions

This research shows that mixed integer linear programming gave the best solution to small size problems. However, large size problems could not be solved using a mathematical model, since there were more variables and limitations. Therefore, we propose differential evolution with a local search that can solve the problem with a solution that is comparable to that of mixed integer linear programming and traditional differential evolution. It has been demonstrated that the differential evolution with a local search obtained optimal solutions. When the problem size was increased, and there are only near-optimal solutions, the differential evolution with a local search was better than the solution by the conventional approach of the ice transportation case study. The DE and the DELS outperformed the current practices in all 27 problems by 0.35-12.13%. The DELS was able to find optimal solutions equivalent to those obtained by an optimization solver, and nearest to the optimal solution.

## 7. Acknowledgements

This work was supported by the Research Unit on System Modeling for Industry (Grant No. SMI.KKU MS 581003), Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Thailand and we thank Mr. Stan Moore and Mr. Ian Thomas for linguistic corrections.

## 8. References

[1] Hartl RF, Hasle G, Janssens GK. Special issue on rich vehicle routing problems. Cent Eur J Oper Res. 2006;14(2):103-4.
[2] Dantzig GB, Ramser JH. The truck dispatching problem. Manag Sci. 1959;6(1):80-91.
[3] Worasan K, Sethanan K, Chaikanha N. Dairy transportation problem with no mixing of raw milk and time windows constraints. Proceedings of the Asia Pacific Industrial Engineering and Management Systems Conference (APIEMS); 2014 Oct 12-15; Jeju Island, Korea; 2014.
[4] Akpinar S. Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. Expert Syst Appl. 2016;61:28-38.
[5] Abdulkader MS, Gajpal Y, Elmekkawy TY. Hybridized ant colony algorithm for the multi compartment vehicle routing problem. Appl Soft Comput. 2015;37:196-203.
[6] Zhang Y, Chen XD. An optimization model for the vehicle routing problem in multi-product frozen food delivery. J Appl Res Tech. 2014;12(2):239-50.
[7] Hsu CI, Hung SF, Li HC. Vehicle routing problem with time-windows for perishable food delivery. J Food Eng. 2007;80(2):465-75.
[8] Osvald A, Stirn LZ. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. J Food Eng. 2008;85(2):285-95.
[9] Jiang J, Ng KM, Poh KL, Teo, KM. Vehicle routing problem with a heterogeneous fleet and time windows. Expert Syst Appl. 2014;41(8):3748-60.
[10] Chamnanlor C, Sethanan K. Bi-objective optimization for reentrant shop scheduling problem. CMU J Nat Sci. 2015;14:447-60.
[11] Jamrus T, Chien CF, Gen M, Sethanan K. Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing. IEEE Trans Semicond Manuf. 2018;31(1):32-41.
[12] Sethanan K, Pitakaso R. Improved differential evolution algorithms for solving generalized assignment problem. Expert Systems With Applications 2016;45:450-9.
[13] Kachitvichyanukul V. Comparison of three evolutionary algorithms: GA, PSO, and DE. Ind Eng Manag Syst. 2012;11(3): 215-23.
[14] Nearchou AC. Meta-heuristics from nature for the loop

layout design problem. Int J Prod Econ. 2006;101(2):312-28.

[15] Wisittipanich W, Hengmeechai P. A multi-objective differential evolution for just-in-time door assignment and truck scheduling in multi-door cross docking problems. Ind Eng Manag Syst. 2015;14(3):299-311.

[16] Xu H, Wen J. Differential evolution algorithm for the optimization of the vehicle routing problem in logistics. The 8th International Conference on Computational Intelligence and Security (CIS 2012); 2012 Nov 17-18; Guangzhou, China. USA: IEEE; 2012. p. 48-51.

[17] Lingjuan HOU, Zhijiang HOU. A novel discrete differential evolution algorithm. Indonesian J Electr Eng Comput Sci. 2013;11(4):1883-8.

[18] Dechampai D, Tanwanichkul L, Sethanan K, Pitakaso R. 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. J Intell Manuf. 2017;28(6):1357-76.

[19] Alinaghian M, Kaviani Z, Khaledan S. A novel heuristic algorithm based on clark and wright algorithm for green vehicle routing problem. Int J Supply Oper Manag. 2015;2(2):784-97.

[20] Ponnambalam SG, Ganesan K. Differential evolution algorithm with local search for capacitated vehicle routing problem. Int J of Bio-Inspired Computation. 2015;7(5):321-42.

[21] Price K, Strom R. Differential evolution a simple evolution strategy for fast optimization. Dobb's J. 1997;22(4):18-24.

[22] Laguna M, Luis J, Velarde G. A search heuristic for just-in-time scheduling in parallel machines. J Intell Manuf. 1991;2(4):253-60.

[23] Laguna M, Barnes JW, Glover F. Intelligent scheduling with tabu search: an application to jobs with linear delay penalties and sequence dependent setup costs and times. Appl Intell. 1993;3(2): 159-72.

[24] De la Cruz JJ, Paternina-Arboleda CD, Cantillo V, Montoya-Torres JR. A two-pheromone trail ant colony system-tabu search approach for the heterogeneous vehicle routing problem with time windows and multiple products. J Heuristics. 2013;19(2):1-20.

[25] Meesuptaweekoon K, Chaovalitwongse P. Dynamic vehicle routing problem with multiple depots. Eng J. 2014;18(4):135-49.

[26] Joshi S, Kaur S. Nearest neighbor insertion algorithm for solving capacitated vehicle routing problem. In: Hoda MN, editor. 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom); 2015 Mar 11-13; New Delhi, India. New Delhi: IEEE; 2015. p. 86-8.