Scheduling the periodic delivery of liquefied petroleum gas tank with time window by using artificial intelligence approaches: An example in Taiwan

Yi-Chih Hsieh\textsuperscript{1}, Peng-Sheng You\textsuperscript{2} and Cheng-Sheng Chen\textsuperscript{1}

\textsuperscript{1}Department of Industrial Management, National Formosa University, Yunlin
\textsuperscript{2}Department of Business Administration, National ChiaYi University, Chia-Yi

Abstract

Introduction In Taiwan, liquefied petroleum gas tank users have to call a gas company to deliver a full liquefied petroleum gas tank when their tank is out of gas. The calls usually congest in the cooking time and the customers have to wait for a long time for a full liquefied petroleum gas tank. Additionally, allocating manpower is difficult for the gas company.

Objectives A strategy of periodic delivery for gas companies was presented to deliver liquefied petroleum gas tanks in advance and charge the gas fee based on the weight of returned tanks. Additionally, a new encoding scheme was proposed and embedded into three evolutionary algorithms to solve the nondeterministic polynomial-hard problem. The objective of the problem is to minimize the total traveling distance of the vehicle such that the delivery efficiency of tanks increases and the waiting time of customer decreases.

Methods A new encoding scheme was presented to convert any random sequence of integers into a solution of the problem and embedded into three evolutionary algorithms, namely, immune algorithm, genetic algorithm, and particle swarm optimization, to solve the delivery problem. Additionally, the encoding scheme can be used to different frequency types of demand based on customers’ requests.

Results Numerical results, including a practical example in Yunlin, Taiwan, were provided to show that the adopted approaches can significantly improve the efficiency of delivery.

Conclusions The periodic delivery strategy and the new encoding scheme can effectively solve the practical problem of liquefied petroleum gas tank in Taiwan.

Corresponding author:
Peng-Sheng You, Department of Business Administration, National ChiaYi University, Chia-Yi 600.
Email: psyuu@mail.ncyu.edu.tw

Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access page (https://us.sagepub.com/en-us/nam/open-access-at-sage).
Keywords
Periodic delivery, liquefied petroleum gas tank, immune algorithm, genetic algorithm, particle swarm optimization

Introduction
Gas is an essential energy in life for cooking food and boiling water. People use two main types of gas, namely, liquefied natural gas (LNG) and liquefied petroleum gas (LPG).

1. LNG: The main components of LNG are 90% methane, 5% ethane, ∼2% propane, and minimal butane and hydrogen. Natural gas produces only water (H₂O) and carbon dioxide after combustion and does not yield harmful gases. It is currently recognized as one of the cleanest commercial energy worldwide.

2. LPG: LPG is the gas in tanks (steel cylinders) used in our houses and it is a mixture of propane and butane extracted from crude oil. LPG itself is actually colorless and odorless. However, once it leaks out, it will become a disaster. Thus, we usually add another odorant to LPG to make people aware and alert.

Usually, LNG is transmitted using pipelines, and LPG is compressed in tanks and delivered using vehicles. In Taiwan, residents in large cities may use LNG directly by pipeline in their houses. However, residents in medium cities or small towns use LPG in their houses due to the setup cost of the pipeline. For the LPG users, they have to call a gas company to deliver a full LPG tank when their tank is empty. The calls usually congest in the cooking time of each day, and the gas company has to deliver many LPG tanks to various areas during the period. Currently, a gas company in Yunlin, Taiwan delivers full LPG tanks to its customers in accordance with the sequence of the order call, that is, first come, first served. Nevertheless, the current delivery strategy presents several drawbacks for gas companies and customers, including the following:

1. **For gas companies:** Customers live in different areas, and their order calls are centralized in the period of cooking time. Thus, effectively allocating manpower and scheduling the delivery tours of vehicles are difficult for gas companies. Usually, two customers live in the same area, but the gas companies have to deliver separately in a short period of time on the basis of their order calls.

2. **For the customers:** Customers call for full LPG tanks only when the old ones are empty or near empty. The order calls are usually at the cooking time of each day; consequently, customers have to wait for a long time to continue their cooking. This situation causes inconvenience. Several customers also complain that their returned LPG tank is incompletely empty, only near empty. However, they have to pay the entire fee for a full LPG tank.

In this study, we present a new strategy for delivering LPG tanks for gas companies to solve the current problems of gas companies and customers simultaneously. In the new strategy, the gas companies can periodically deliver full LPG tanks to their customers in advance and charge the LPG fee on the basis of the weight of returned tanks.
That is, customers will receive full LPG tanks in advance, although their tanks are not empty. The main advantages of the new strategy are multiple, including the following:

1. The delivery efficiency increases because the gas companies can efficiently allocate their manpower and schedule the delivery path of vehicles in advance.
2. The waiting time of customers decreases because they can obtain a full tank before the old one is completely empty. The LPG fee is also based on the weight of the returned tank, which will reduce customer complaints.

In the following, we briefly review the related research of the delivery problem in the “Literature review” section and then present the model in the “Periodic delivery problem of LPG tank (PDP-LT)” section. In the “Approaches” section, approaches are introduced. Finally, numerical results, discussion, and conclusions are presented in the “Test problems,” “Discussions,” and “Conclusions” sections, respectively. The main structure of this study is illustrated in Figure 1.

**Literature review**

The considered PDP-LT is an extension of the typical vehicle routing problem (VRP). The VRP was proposed by Dantzig and Ramser¹; vehicles are used to visit each customer once, such that the total traveling distance is minimized. The VRP has many variants,
including capacitated VRP, VRP with time windows, multiple-depot VRP, VRP with pickup and delivery (VRPPD), split delivery VRP, stochastic VRP, periodic VRP (PVRP), VRP with backhauls, etc. The considered PDP-LT is more related to VRPPD and PVRP, however, there exist some significant differences between the PDP-LT and these two problems. We briefly review as follows:

1. VRPPD extends VRP and assumes that the vehicle has to deliver goods from the depot to customers or/and pick up goods from customers to the depot. This assumption is similar to that of the considered PDP-LT in which the vehicle has to deliver full LPG tanks to customers and pick up empty or near-empty LPG tanks from customers to the gas company. However, in practice, customers of PDP-LT usually have periodic demands for a fixed number of full LPG tanks, which are not involved in the VRPPD. The VRPPD is a nondeterministic polynomial (NP)-hard problem (Furtado et al.\textsuperscript{2}) and many approaches have been proposed to solve it. For example, Rodrigues et al.\textsuperscript{3} presented a mixed integer programming model and two heuristics to solve the VRPPD in a maritime oil transportation system. Furtado et al.\textsuperscript{2} proposed an integer mathematical programming model with a two-index formulation to solve VRPPD. Karami et al.\textsuperscript{4} investigated a dynamic VRPPD with time windows and urgent requests. They proposed a two-step heuristic with the cheapest insertion and local search to solve the problem. To shorten this article, we refer to Parragh et al.\textsuperscript{5,6} for the variants, models, and approaches of VRPPD.

2. PVRP was first introduced by Beltrami and Bodin\textsuperscript{7} in which multiple vehicles with limited capacity are used to visit each customer for a given number of times in the planning time horizon, such that the total traveling distance is minimized. If the number of visits for each customer is one, then the PVRP leads to the VRP. Several variants of PVRP have been proposed with various assumptions and constraints. They include multiple-depot PVRP, PVRP with time windows, PVRP with service choice, and PVRP with consistency. Additional references include Parthanadee and Logendran,\textsuperscript{8} Francis et al.,\textsuperscript{9} An et al.,\textsuperscript{10} and Rodríguez-Martín et al.\textsuperscript{11} The PVRP and its variants have attracted considerable attention in recent years due to their numerous practical applications, such as courier services, maintenance, replenishment, waste collection, interlibrary loan service, and lottery retailer visit. The related references are Parthanadee and Logendran,\textsuperscript{8} Blakely et al.,\textsuperscript{12} Jang et al.,\textsuperscript{13} Gaur and Fisher,\textsuperscript{14} Claassen and Hendriks,\textsuperscript{15} Mourgaya and Vanderbeck,\textsuperscript{16} Francis et al.,\textsuperscript{17} Teixeira et al.,\textsuperscript{18} and Baptista et al.\textsuperscript{19} Different approaches have been proposed in the past few years to solve the PVRP and its variants. The examples are as follows:

(a) Classic heuristics, for example, Russell and Gribbin\textsuperscript{20} and Gaudioso and Paletta.\textsuperscript{21}

(b) Metaheuristics, for example, Cordeau et al.,\textsuperscript{22} Drummond et al.,\textsuperscript{23} Hemmelmayr et al.,\textsuperscript{24} and Tenahua et al.\textsuperscript{25}

(c) Mathematical programming approaches, for example, Francis et al.,\textsuperscript{9} Mourgaya and Vanderbeck,\textsuperscript{16} and Francis et al.\textsuperscript{17}

Further references of the PVRP and its variants, approaches, and models are the excellent survey papers by Francis et al.\textsuperscript{17} and Campbell and Wilson.\textsuperscript{26}
The considered PDP-LT is related to the PVRP, but the following notes are provided:

1. The considered PDP-LT differs from the PVRP because the delivery frequency of demand for a customer is given in the PDP-LT, that is, the time interval of two consecutive visits is fixed for a customer in the planning time horizon. In the PVRP, the number of visits for a customer is given, but the interval of two consecutive visits is not fixed in the planning time horizon.

2. The considered PDP-LT is similar but differs from the periodic home health-care-scheduling problem (HHSP) investigated by An et al. In the HHSP, several nurses are scheduled to provide home health-care services periodically in the planning time horizon, such that the total traveling distance is minimized. Although the considered PDP-LT and HHSP are also NP-hard problems, the PDP-LT differs from the HHSP because of the following reasons:
   (a) For the assumption aspect: The PDP-LT considers the time window of the customer, which is not considered in the HHSP.
   (b) For the method aspect: The HHSP proposes a two-phase heuristic to construct a feasible solution. Specifically, phase-1 heuristic is used to construct partial schedules for patients with a service frequency of once for 2 and 3 days, and phase-2 heuristic is used to include patients with service frequency of once for 6 and 12 days into the schedules by using phase-1 heuristic. However, the current paper presents a new encoding scheme to convert any random sequence of integers into a solution to the problem directly.

Next, we summarize the main objectives and contributions of this study as follows:

1. Proposing a new strategy for solving the practical PDP-LT in Yunlin, Taiwan. As shown, the new strategy can improve the delivery efficiency of a vehicle and reduce the waiting time of customers.

2. Presenting a new encoding scheme to convert any random sequence of integers into a solution of the PDP-LT directly, which differs from the typical two-phase methods proposed by several researchers. We embed the new encoding scheme into three artificial intelligence algorithms, namely, immune algorithm (IA), genetic algorithm (GA), and particle swarm optimization (PSO), to solve the PDP-LT. The presented encoding scheme can also be used for various delivery frequency types of demand based on customers’ requests.

3. Presenting the numerical results of a practical example in Yunlin, Taiwan, and extended examples with time windows of customers. We compare and discuss the numerical results to show the effectiveness and efficiency of the adopted approaches.

**Periodic delivery problem of LPG tank (PDP-LT)**

The notation (see Appendix 1), assumptions, and mathematical models of the PDP-LT are as follows.
Assumptions

1. The network comprises \( n \) customers and the demand quantity and frequency of each customer for each visit are given. However, the frequency type of delivery \( k, k \in F \), for each customer, has to be decided.
2. Three possible time windows of delivery for each customer, namely, a.m., p.m., and a.m./p.m., are given.
3. A single vehicle is available to deliver and its capacity is given.
4. For each day, the vehicle departs from the depot to deliver and then goes back to the depot for replenishment. Finally, the vehicle has to go back to the depot when all customers scheduled have been served for each time window of the day.
5. The service time of each customer is ignored.
6. The PDP-LT aims to minimize the total traveling distance of the vehicle for all customers during the considered planning time horizon \( T \). In this study, we set the objective as \( \sum_t \sum_w R_t(w) + P_t(w) \), that is, the sum of the total traveling distance of the vehicle and the penalty if the vehicle cannot finish the scheduled delivery in the time window.

Mathematical model

The mathematical model of PDP-LT shown below is similar to that of An et al.\(^{10}\) Nevertheless, the PDP-LT considers the time window constraint for each customer, which is ignored in their model:

\[
\text{Min} \quad \sum_t \sum_w R_t(w) + P_t(w) \\
\text{st} \quad \sum_j x_{ij}(w) = y_{ii}(w), \quad \forall \; i \in C^k, \; k \in F, \; 1 \leq t \leq T - k, \; w \in W
\]  

\[
\sum_{i \in S} \sum_{j \in S} \sum_w x_{ij}(w) \leq |S| - 1, \quad S \subseteq C, \; S \neq \emptyset, \; S \neq C, \; 1 \leq t \leq T - k, \; k \in F
\]

\[
\sum_{ij \in V_{og}(w)} x_{ij}(w) q_i \leq Q, \quad 1 \leq g \leq h_i(w), \; 1 \leq t \leq T - k, \; k \in F, \; w \in W
\]

\[
x_{ij}(w) \in \{0, 1\}, \; y_{ij}(w) \in \{0, 1\}, \quad \forall i \neq j, \; 1 \leq t \leq T - k, \; w \in W
\]
for customers. That is, the demand frequency $k$ of customers must be satisfied, $k \in F = \{2, 3, 6\}$. Since constraints (1) and (2) only ensure that all customers in the time window $w$ of each day should be served. Thus, sometimes the invalid sub tour solution, in which the routes of customers are not connected, will occur. Thus, constraint (3) is used to eliminate all possible sub tours of the vehicle, that is, delete all infeasible delivery tours. Constraint (4) is the vehicle capacity constraint. Constraint (5) is the range for all binary variables.

**Approaches**

**Encoding scheme and example**

For the PDP-LT, we present a new encoding scheme to convert a random permutation of \{1, 2, ..., $N$\} with a module operator into a solution of the problem, where $N$ is the total number of visits for customers in the planning time horizon $T$. Specifically, in the new encoding scheme, we adopt a module operator to decide the frequency type of delivery for each customer. We also use the module operator to determine the time window for each customer. The following example is adopted to show the main steps of the new encoding scheme.

**Example.** The assumptions of the example are as follows:

1. Ten customers are to be served, in which $C^2 = \{18, 15, 43, 48, 23, 29\}$, $C^3 = \{12, 31\}$, and $C^6 = \{5, 9\}$.
2. The depot is located at node 22 (D).
3. The available delivery time window for each customer is shown in Table 1.
4. The considered time horizon is $T = 6$ days.

Our main encoding schemes are illustrated using the following steps:

1. Considering that $C^2 = \{18, 15, 43, 48, 23, 29\}$, $C^3 = \{12, 31\}$, $C^6 = \{5, 9\}$, and $N = 6 \times 2 + 2 \times 3 + 2 \times 6 = 30$, we may use a random permutation of integer sequence 1 to 30 to represent a solution of the PDP-LT.
2. The random sequence is supposed as $R = 21, 1, 9, 13, 14, 4, 3, 11, 19, 28, 18, 22, 17, 7, 25, 10, 29, 20, 15, 30, 23, 26, 16, 24, 5, 27, 2, 12, 6, 8$.
3. As shown in Table 1, node 18 is in $C^2$; hence, we assign the first two random numbers in $R$, that is, 21 and 1, to this node. The regular sequence of 1 to 30 is added as those of $I$ into column 5 of Table 1.
4. Node 18 is in $C^2$ and has three possible frequency types of delivery, namely, 100100, 010010, and 001001. Thus, we may compute mod $(21 + 1, 3) + 1 = 2$. That is, we select the frequency type of 010010 for node 18 and assign its corresponding random numbers 21 and 1 to days 2 and 5, respectively, as those in Table 1.
5. Similar steps are repeated for the other nodes of customers. We have the random assignment of days 1 to 6, which is shown in Table 1. Based on Table 1, we may construct the delivery sequence for the vehicle. For example, on day 1, the order...
Table 1. The encoding scheme for the example.

| Node | Time window | R | I | \( \text{mod} + 1 \) | Selected type | D1 | D2 | D3 | D4 | D5 | D6 |
|------|-------------|---|---|----------------|---------------|----|----|----|----|----|----|
| 18   | a.m.        | 21, 1 | 1, 2 | \( \text{mod}(21 + 1, 3) + 1 = 2 \) | 010010 | 0 | 21 | 0 | 0 | 0 | 1 |
| 15   | a.m.        | 9, 13 | 3, 4 | \( \text{mod}(9 + 3, 3) + 1 = 1 \) | 100100 | 9 | 0 | 0 | 13 | 0 | 0 |
| 43   | a.m./p.m.   | 14, 4 | 5, 6 | \( \text{mod}(14 + 5, 3) + 1 = 2 \) | 010010 | 0 | 14 | 0 | 0 | 0 | 4 |
| 5    | a.m./p.m.   | 3, 11, 19, 28, 18, 22 | 7, 8, 9, 10, 11, 12 | \( \text{mod}(3 + 7, 1) + 1 = 1 \) | 111111 | 3 | 11 | 19 | 28 | 18 | 22 |
| 48   | p.m.        | 17, 7 | 13, 14 | \( \text{mod}(17 + 13, 3) + 1 = 1 \) | 100100 | 17 | 0 | 0 | 7 | 0 | 0 |
| 23   | p.m.        | 25, 10 | 15, 16 | \( \text{mod}(25 + 15, 3) + 1 = 2 \) | 010010 | 0 | 25 | 0 | 0 | 0 | 10 |
| 12   | p.m.        | 29, 20, 15 | 17, 18, 19 | \( \text{mod}(29 + 17, 2) + 1 = 1 \) | 101010 | 29 | 0 | 20 | 0 | 15 | 0 |
| 31   | p.m.        | 30, 23, 26 | 20, 21, 22 | \( \text{mod}(30 + 20, 2) + 1 = 1 \) | 101010 | 30 | 0 | 23 | 0 | 26 | 0 |
| 9    | a.m.        | 16, 24, 5, 27, 2, 23, 24, 25, 26, 12 | 27, 28 | \( \text{mod}(16 + 23, 1) + 1 = 1 \) | 111111 | 16 | 24 | 5 | 27 | 2 | 12 |
| 19   | a.m./p.m.   | 6, 8 | 29, 30 | \( \text{mod}(6 + 29, 3) + 1 = 3 \) | 001001 | 0 | 0 | 6 | 0 | 0 | 8 |
6. Similarly, we obtain the entire delivery sequence of 6 days for the vehicle, as shown below.

Day 1: (D)→5→15→9→48→12→31→(D)
Day 2: (D)→5→43→18→9→23→(D)
Day 3: (D)→9→19→5→12→31→(D)
Day 4: (D)→48→15→9→5→(D)
Day 5: (D)→18→9→43→23→12→5→31→(D)
Day 6: (D)→19→9→5→(D)

7. We adopt the following procedure to extend the encoding scheme for various delivery windows of customers. Day 1 is regarded as an example. The delivery order of node for day 1 is (D)→5→15→9→48→12→31→(D), and its corresponding random sequence is 3, 9, 16, 17, 29, and 30. The first customer is node 5 with time window a.m./p.m., that is, two options of time window for this node. Thus, we may compute \( \text{mod} (3 + 9, 2) + 1 = 1 \), which implies that node 5 is to be served in a.m. (option 1) on day 1, where 3 and 9 are the first two numbers of the order sequence of day 1. Similarly, the second customer is node 15 with time window a.m., that is, one option for a time window. Thus, we may compute \( \text{mod} (9 + 16, 1) + 1 = 1 \), which implies that node 15 is to be served in a.m. (option 1) on day 1, where 9 and 16 are the second and third numbers of the order sequence of day 1. Repeating the same procedure to all nodes, we finally have the following:

Day 1: a.m.: (D)→5→15→9→31→(D)
   p.m.: (D)→48→12→(D)
Day 2: a.m.: (D)→18→9→(D)
   p.m.: (D)→5→43→23→(D)
Day 3: a.m.: (D)→9→23→(D)
   p.m.: (D)→19→5→12→(D)
Day 4: a.m.: (D)→15→9→(D)
   p.m.: (D)→48→5→(D)
Day 5: a.m.: (D)→18→9→43→5→(D)
   p.m.: (D)→23→12→31→(D)
Day 6: a.m.: (D)→19→9→5→(D)
   p.m.: No tour

Accordingly, for the random permutation \( R \), we may compute the objective, that is, the total traveling distance, based on the vehicle capacity and the above assignment for each day. Since the obtained solution in each time window may violate the upper limit of the traveling distance of the vehicle, that is, \( U_t(w) \). Thus, as shown in the mathematical model, we add a penalty to the objective if the vehicle cannot finish the scheduled delivery in the time window. Additionally, this example shows that the proposed encoding procedure can convert a random sequence of \( 1 \) to \( N \), that is, a permutation of \( \{1, 2, \ldots, N\} \), to present a solution vector of the PDP-LT. Consequently, typical crossover and mutation operators of IA and GA are shown in Figure 2 can be used to produce new
solutions (individuals). To shorten this article, we omit the detailed descriptions of operators in IA and GA.

The above encoding scheme can be further extended to various demand frequency types based on customers’ requests. For example, three different customers’ requests of demand and delivery types are as follows:

Customer 1: Deliver twice per week, no delivery on Friday and Saturday, must be served once on Thursday. Then the possible delivery types are 100100, 010100, and 001100.
Customer 2: Deliver twice per week, no delivery on Monday and Wednesday, must be served once on Friday. Then the possible delivery types are: 010010, 001110, and 000011.
Customer 3: Deliver twice per week, must be served on Tuesday and Saturday. Then the delivery type is 010001.

Thus, in Table 1, one may use a random sequence of 1 to 6 to represent a solution for the problem and the procedure of assignment of the time window is similar to step (7) of the above example. This extension also implies that the proposed encoding scheme can also be used to convert a random sequence to a solution of the problem based on the demand frequency type of customers’ requests even when the demand frequency type is not periodic.

### Three algorithms

In this subsection, we introduce the three algorithms used to solve the PDP-LP with the embedding of the new proposed encoding scheme in the “Encoding scheme and example” section.

In the past, many researchers were devoted to finding global optimal solutions for different NP-hard optimization problems by using mathematical programming approaches via the branch-and-cut method. However, in the past few years, instead of obtaining a global optimum, researchers are practical and interested in obtaining a near-global optimum within a reasonable time. Many types of artificial intelligence evolutionary algorithms have recently been widely proposed to solve different complicated optimization problems.

![Figure 2. Solution presentation, crossover and mutation of IA and GA. GA: genetic algorithm. IA: immune algorithm.](image-url)
For example, Mirjalili and Lewis\textsuperscript{27} surveyed more than 50 novel optimization algorithms based on nature or animal behavior. They also classified recent novel optimization algorithms into three main groups, namely, evolution-, physics-, and swarm-based methods. Among all artificial intelligence algorithms, GA, proposed by Holland\textsuperscript{28} and PSO, proposed by Kennedy and Eberhart\textsuperscript{29} may be the most popular and adopted methods to solve different complicated optimization problems due to their considerable success in the literature.

IA, an immunity-based evolutionary algorithm, has also attracted much attention due to its variety in solution (Jiang et al.\textsuperscript{30}). Unlike PSO, a swarm-based method, IA and GA are evolution-based algorithms. IA is similar to GA in most of the evolution mechanisms, such as crossover and mutation. Nonetheless, IA possesses a unique memory mechanism that is not involved in GA. Specifically, IA uses the memory mechanism to delete excessively similar solutions, although they are excellent in the objective value; hence, the variety of solutions can be kept to improve the objective. Consequently, IA could have a higher probability to outperform GA in solving various optimization problems (Hsieh et al.\textsuperscript{31} and Jiang et al.\textsuperscript{30}). In this study, we adopt IA to solve the PDP-LT. We also provide, discuss, and compare the numerical results of IA with those of GA and PSO. The main steps of IA, GA, and PSO are shown in Figures 3 to 5, respectively, and for their detailed descriptions refer to Hsieh et al.\textsuperscript{31}, Holland\textsuperscript{28} and Kennedy and Eberhart\textsuperscript{29}. Note that the main steps of the three algorithms are similar to those in the literature. However, in this paper, we embed the new proposed encoding scheme into the three algorithms to increase their effectiveness.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{IA_diagram.png}
\caption{Main steps of immune algorithm (IA; Hsieh et al.\textsuperscript{30}).}
\end{figure}
Test problems

Two main test problems are considered in this study. The first one is an example in Yunlin, Taiwan, which is illustrated in Figure 6, and the second one is an extension of the first one. The two main test problems are summarized as follows.

1. Test problem 1 (30 of 60 customers, no time window). This problem is a real example of a gas company in Yunlin, Taiwan with 30 demand customers and a vehicle. The original data of the company for demand customers (nodes) and demand calls are shown in Appendix 2, and the demand quantity for each customer is shown in Appendix 3. Based on Appendixes 2 and 3, we approximate the demand cycle and quantity to set $C_k$ and assume the time window of customers’ based on their preference. For example, Appendix 3 shows that the demand calls of node 143 are on days 2, 3, 4, 5, and 6 in weeks 1 and 2, days 2, 4, 5, and 6 in week 3, days 2, 3, 5, and 6 in week 4, respectively. Appendix 3 also shows that the demand for each call for node 143 is one. Thus we set the demand frequency for node 143 is $k = 6$ and its demand is 1. The corresponding data of the PDP-LT are summarized in Table 2. In addition to the case of 30 customers, we also extend the case to 60 customers. Thus, as shown in Table 2, four cases of PDP-LP are tested with node numbers 30 and 60, and the vehicle capacities are 15 and 30.

2. Test problem 2 (60 customers, time window). Test problem 2 is the extended case of the gas company in Yunlin, Taiwan, with 60 demand customers and adds the time windows for customers. The vehicle capacities are assumed as 15 and 30 tanks. The corresponding data of Test problem 2 are shown in Table 3.
For Test problems 1 and 2, we perform 30 trials by using GA, IA, and PSO. The best, worst, and average of solutions, central processing unit (CPU) times, and convergence iterations of the algorithms in 30 trials are summarized in Tables 4 and 5 and Figure 7 for Test problems 1 and 2, respectively. For IA and GA, we set: population = 300, affinity = 0.25, crossover = 0.6878, mutation = 0.3, maximum generations = 20,000, and maximum no. of reproduction of each chromosome = 7. For PSO, we set \((c_1, c_2) = 0.4, 0.6\), population = 500, and maximum generations = 20,000. The three adopted algorithms are set to stop if no improvement occurs for 1000 consecutive iterations. Our algorithm programs are coded in MATLAB R2015, and all results are computed using Intel-Pentium IV 4 CPU 3.0 GHz PC.

**Discussions**

From Table 4, Figure 7(a) and (b), we have the following findings for Test problem 1:

1. For the real example of a gas company in Taiwan, that is, 30 customers with vehicle capacity = 15, Table 6 shows that the total delivery distance of the old strategy of the gas company, that is, first call, first served, for the 4 weeks of a month is 1463.65, 1401.5, 1321.4, and 1262.65, and the average delivery distance is 1362.3. The total number of delivery for 6 days (a week) is 41, 50, 45, and 46 with an average of 45.5 times. However, IA reports that the best objective value is 582.3 for the same case. That is, the new delivery strategy can effectively reduce the total delivery distance and the total number of tours for the vehicle.

![Figure 5](image_url). Main steps of particle swarm optimization (PSO; Kennedy and Eberhart).
2. The objective value decreases with the increase in vehicle capacity. For example, IA reports that the best objective value is 582.3 for the case with a capacity of 15 and 507.0 for the case with a capacity of 30. This observation implies that increasing the vehicle capacity will decrease the objective, that is, the total traveling distance of the vehicle. We have similar findings for GA and PSO.

3. The standard deviation of the objective value decreases with the increase in the number of customers. For example, the standard deviations of objective value
by using IA are 17.9 and 15.1 for Test problem 1 with vehicle capacities of 15 and 30, respectively. Nevertheless, we have no such a finding for GA and PSO.

4. The average convergence iteration seems stable for all test problems, but no significant difference is determined for various capacities of the vehicle. For example, the average convergence iteration by using IA is 2276.6 for Test problem 1 with a vehicle capacity of 15, and it is 2141.1 for the case with a vehicle capacity of

| Customers | Type | Node (demand) |
|-----------|------|---------------|
| 30        | C^6  | 143(1), 145(1), 147(1), 152(1), 154(2), 156(1), 158(2), 160(3), 165(2), 167(2) |
|           | C^3  | 146(2), 148(3), 149(2), 155(1), 159(1), 161(2), 162(1), 166(2), 168(3), 169(2), 172(2) |
|           | C^2  | 144(2), 150(1), 151(1), 153(1), 157(1), 163(1), 164(1), 170(2), 171(1) |
|           | C^6  | 143(2), 145(2), 147(2), 152(2), 154(4), 156(2), 158(4), 160(6), 165(4), 167(4), 40(2), 130(4), 116(4), 62(2), 106(2), 92(2), 33(2), 53(2) |
| 60        | C^3  | 144(4), 150(2), 151(2), 153(2), 157(2), 163(2), 164(2), 170(4), 171(2), 110(4), 24(2), 96(2), 21(2), 30(4), 36(4), 59(4), 15(2), 47(4), 3(2) |
|           | C^2  | 146(4), 148(6), 149(4), 155(2), 159(2), 161(4), 162(2), 166(4), 168(6), 169(4), 172(4), 54(2), 64(2), 97(2), 88(4), 49(4), 82(4), 25(4), 62(2), 42(2), 27(2), 19(2), 18(2) |

| Table 2. Test problem 1 (no time window). |
| Customers | Type | Node (demand) |
|-----------|------|---------------|
|           | C^6  | 143(1), 145(1), 147(1), 152(1), 154(2), 156(1), 158(2), 160(3), 165(2), 167(2) |
|           | C^3  | 146(2), 148(3), 149(2), 155(1), 159(1), 161(2), 162(1), 166(2), 168(3), 169(2), 172(2) |
|           | C^2  | 144(2), 150(1), 151(1), 153(1), 157(1), 163(1), 164(1), 170(2), 171(1) |
|           | C^6  | 143(2), 145(2), 147(2), 152(2), 154(4), 156(2), 158(4), 160(6), 165(4), 167(4), 40(2), 130(4), 116(4), 62(2), 106(2), 92(2), 33(2), 53(2) |

| Table 3. Test problem 2 (60 demand nodes with time window). |
| Customers | Type | Time window |
|-----------|------|-------------|
|           | C^6  | a.m. 147(2) 152(2) 154(4) 156(2) d. 62(2) 92(2) p.m. 143(2) 160(6) 167(4) 40(2) 130(4) 53(2) a.m./p.m. 145(2) 158(4) 165(4) 116(4) 106(2) 33(2) |
| 60        | C^3  | a.m. 150(2) 157(2) 164(2) 170(4) 171(2) 15(2) p.m. 163(2) 96(2) 30(4) 36(4) 24(2) 21(2) 59(4) 47(4) 144(4) 151(2) 153(2) 110(4) a.m./p.m. 3(2) |
|           | C^2  | a.m. 149(4) 162(2) 49(4) 62(2) 42(2) p.m. 159(2) 161(4) 64(2) 82(4) 146(4) 148(6) 155(2) 166(4) 168(6) 169(4) 172(4) 54(2) 64(2) 97(2) 88(4) 49(4) 82(4) 25(4) 62(2) 42(2) 27(2) 19(2) 18(2) 54(2) 25(4) 27(2) 19(2) 18(2) 54(2) 25(4) 27(2) 19(2) 18(2)
Table 4. Numerical results of test problem 1 for various algorithms under different capacity of vehicle (no time window).

| No. of customers | Vehicle capacity | GA          | IA          | PSO         |
|------------------|------------------|-------------|-------------|-------------|
|                  |                  | Min | Avg | Max | Std | Min | Avg | Max | Std | Min | Avg | Max | Std | Min | Avg | Max | Std |
| 15               |                  | obj  | 592.5 | 629.5 | 655.0 | 15.3 | 582.3 | 621.3 | 660.0 | 17.9 | 781.3 | 852.1 | 924.5 | 36.2 |
|                  |                  | cpu  | 4570.3 | 6401.6 | 10919.0 | 1378.7 | 5245.4 | 7425.2 | 12190.0 | 1823.5 | 1566.2 | 3048.0 | 4790.4 | 782.1 |
|                  |                  | iter | 936.0 | 1700.1 | 3586.0 | 575.2 | 1062.0 | 2276.6 | 4399.0 | 838.5 | 968.0 | 2824.7 | 5012.0 | 980.3 |
| 30               |                  | obj  | 523.6 | 557.8 | 614.7 | 22.4 | 507.0 | 541.0 | 572.2 | 15.1 | 721.0 | 793.4 | 868.9 | 38.7 |
|                  |                  | cpu  | 3972.2 | 6254.4 | 9500.8 | 1392.9 | 4273.6 | 7247.8 | 12138.0 | 2049.1 | 1921.0 | 2946.6 | 4465.0 | 631.2 |
|                  |                  | iter | 698.0 | 1642.2 | 2982.0 | 609.0 | 811.0 | 2141.1 | 4072.0 | 884.6 | 1450.0 | 2742.7 | 4641.0 | 797.9 |
| 60               |                  | obj  | 1058.5 | 1092.0 | 1130.0 | 20.7 | 1018.1 | 1084.6 | 1147.1 | 29.0 | 1636.1 | 2013.6 | 2150.4 | 99.7 |
|                  |                  | cpu  | 9284.2 | 16681.4 | 24468.0 | 4592.2 | 12672.0 | 24245.4 | 40601.0 | 7222.1 | 1045.0 | 2680.6 | 7462.8 | 1506.4 |
|                  |                  | iter | 2737.0 | 5393.9 | 7765.0 | 1660.9 | 1644.0 | 5336.2 | 11969.0 | 2302.7 | 308.0 | 2317.6 | 8267.0 | 1798.3 |
| 30               |                  | obj  | 917.9 | 1000.2 | 1546.5 | 150.7 | 883.7 | 948.3 | 995.2 | 27.4 | 1640.9 | 1861.1 | 2023.8 | 109.5 |
|                  |                  | cpu  | 2766.0 | 14290.0 | 23191.9 | 4648.8 | 15086.0 | 19799.0 | 28107.6 | 3714.1 | 1152.1 | 3472.0 | 8114.2 | 1924.8 |
|                  |                  | iter | 126.0 | 4794.7 | 8226.0 | 1865.1 | 7082.0 | 9733.3 | 13475.0 | 1691.1 | 323.0 | 2575.9 | 7048.0 | 1862.7 |

GA: genetic algorithm; IA: immune algorithm; PSO: particle swarm optimization.
Table 5. Numerical results of test problem 2 for various algorithms under different capacity of vehicle (with time window).

| No. of Customers | Vehicle capacity | GA | IA | PSO |
|------------------|------------------|----|----|-----|
|                  | Min             | Avg | Max | Std  | Min             | Avg      | Max   | Std  |
| 15               | 1138.3          | 1180.6 | 1254.2 | 26.1 | 1134.7          | 1178.1  | 1227.7 | 23.0 |
|                  | obj             | cpu  | iter |      | 1138.3          | 1180.6  | 1254.2 | 26.1 |
| 60               | 977.6           | 1025.2 | 1076.0 | 23.4 | 960.2           | 1022.6  | 1070.3 | 23.1 |
|                  | obj             | cpu  | iter |      | 977.6           | 1025.2  | 1076.0 | 23.4 |
| 30               | 8950.5          | 15913.1 | 27895.5 | 4582.2 | 14984.5         | 24061.2 | 35715.0 | 6263.2 |
|                  | cpu             | iter |      |      | 8950.5          | 15913.1 | 27895.5 | 4582.2 |

GA: genetic algorithm; IA: immune algorithm; PSO: particle swarm optimization.
30. We have similar findings for GA and PSO. Figure 8(a) also illustrates the findings of average convergence iteration for the three algorithms. Therefore, the three algorithms are stable in convergence iteration for solving the PDP-LT.

5. The CPU time of the algorithm increases with the increase in the number of customers. For example, the CPU time by using IA is 7425.2 for Test problem 1 with 30 customers and a vehicle capacity of 15, whereas it is 24,245.4 with 60 customers and a vehicle capacity of 15. This observation implies that more customers will lead to a more complicated PTP-LT. We have similar findings for GA and PSO.

6. Among the three artificial intelligence algorithms, IA performs better than GA, and GA outperforms PSO. For example, the best objective value of Test

![Figure 7. Best solution of three algorithms for two test problems.](image)
problem 1 with a vehicle capacity of 15 is 582.3, 592.5, and 781.3 by using IA, GA, and PSO, respectively. Since the main difference between IA and GA is its memory mechanism, the results also imply that the memory mechanism of IA can be used to improve the objective.

7. Among the three artificial intelligence algorithms, PSO is faster than GA and GA is faster than IA. For example, the average CPU time for Test problem 1 with a vehicle capacity of 15 is 3048.0, 6401.6, and 7425.2 by using PSO, GA, and IA, respectively. Figure 8(b) also illustrates the findings of CPU time for the three algorithms.

From Table 5 and Figure 7(c), we have the following observations for Test problem 2:

1. With the constraint of the time window for each customer, the objective value, that is, the total delivery distance, increases for all cases of Test problem 2. For example, IA reports that the best objective value is 883.7 for Test problem 1 (no time window) with 60 customers and a vehicle capacity of 30, whereas it is 960.2 for Test problem 2 (with time window). We have similar findings for GA and PSO. The result implies that the time window constraint of the customer will reduce the feasible region of PDP-LT and result in an increased objective value.

2. For all cases in Test problem 2, IA outperforms GA, and GA outperforms PSO. For example, the best objective is 1134.7 for the case with a vehicle capacity of 15, whereas it is 1138.3 and 2003.8 for GA and PSO, respectively.

3. For all test problems, PSO is faster than GA, and GA is faster than IA.

4. For all test problems, IA needs more iteration to converge than GA and PSO.

We test the following hypotheses by using the numerical results of 30 trials to analyze the performance of the three adopted algorithms in this paper:

\[ H_0: \text{Avg}(A) = \text{Avg}(B) \]

\[ H_1: \text{Avg}(A) \neq \text{Avg}(B) \]

where \( \text{Avg}(A) \) and \( \text{Avg}(B) \) denote the average objective, that is, the total traveling distance of the vehicle, by using algorithms \( A \) and \( B \), respectively, where \( A, B \in \{ \text{IA, GA, PSO} \} \), and \( A \neq B \). The corresponding \( p \)-values of the statistical hypotheses are summarized in Table 7, which shows the following results:

Table 6. The current solution of gas company.

| Week | 1       | 2       | 3       | 4       | Average |
|------|---------|---------|---------|---------|---------|
| Total delivery distance | 1463.65 | 1401.50 | 1321.40 | 1262.65 | 1362.3  |
| No. of delivery tours for 6 days | 51 \((=8+10+7+7+10+9)\) | 50 \((=9+8+8+8+9+8)\) | 45 \((=10+5+6+8+7+9)\) | 46 \((=8+6+8+8+8+8)\) | 45.5    |
1. For Test problem 1, except for the case with a vehicle capacity of 15, IA outperforms GA, with a \( p \)-value of 0.063. However, for all cases, IA outperforms PSO and GA outperforms PSO.

2. For Test problem 2, IA and GA have no significant difference for the two cases with capacities of 15 and 30. However, for all cases, IA outperforms PSO and GA outperforms PSO.

**Conclusions**

This paper presented a new periodic delivery strategy for gas companies to deliver LPG tanks in advance although their LPG tanks are not empty. The strategy can improve the efficiency of delivery for gas companies and decrease the waiting time of receiving a new full LPG tank for customers. In this paper, a novel encoding scheme was proposed to convert any random integer sequence into a solution of the PDP-LT directly. Three evolutionary algorithms, including IA, GA, and PSO, with the encoding scheme had been applied to solve the PDP-LT. The numerical results of practical examples in Taiwan show that the three algorithms can significantly reduce the total delivery distance and IA is superior to the other two algorithms. Additionally, the presented encoding scheme can be further extended to various demand frequency types based on customers’ requests even if their frequency types are not periodic.
In the future, one may consider a more generalized PDP-LT with the other practical constraints, for example, multiple vehicles, multiple depots, and customers’ emergence requests. Additionally, the other encoding schemes and evolutionary algorithms can be developed to solve the PDP-LT and compare their efficiency and effectiveness.

Declaration of conflicting interests
The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was partially supported by Ministry of Science and Technology, Taiwan, under Grant No. MOST 108-2221-E-150-002 and MOST 109-2221-E-150-027.

ORCID iD
Yi-Chih Hsieh  https://orcid.org/0000-0002-6411-4731

References
1. Dantzig GB and Ramser JH. The truck dispatching problem. Manage Sci 1959; 6: 80–91.
2. Furtado MGS, Munari P and Morabito R. Pickup and delivery problem with time windows: a new compact two-index formulation. Oper Res Lett 2017; 45: 334–341.
3. Rodrigues VP, Morabito R, Yamashita D, et al. Ship routing with pickup and delivery for a maritime Oil transportation system: mIP model and heuristics. System 2016; 4: 31.
4. Karami F, Vancroonenburg W and Berghe GV. A periodic optimization approach to dynamic pickup and delivery problems with time windows. J Sched 2020; 23: 711–731.
5. Parragh SN, Doerner KF and Hartl RF. A survey on pickup and delivery problems: Part I: Transportation between customers and depot. J Betr.wirtsch 2008; 58: 21–51.
6. Parragh SN, Doerner KF and Hartl RF. A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. J Betr.wirtsch 2008; 58: 81–117.
7. Beltrami EJ and Bodin LD. Networks and vehicle routing for municipal waste collection. Networks 1974; 4: 65–94.
8. Parthanadee P and Logendran R. Periodic product distribution from multi-depots under limited supplies. *IIE Trans* 2006; 38: 1009–1026.
9. Francis PM, Smilowitz KR and Tzur M. The period vehicle routing problem with service choice. *Transp Sci* 2006; 40: 439–454.
10. An YJ, Kim YD, Jeong BJ, et al. Scheduling healthcare services in a home healthcare system. *J Oper Res Soc* 2012; 63: 1589–1599.
11. Rodríguez-Martín I, Salazar-González JJ and Yaman H. The periodic vehicle routing problem with driver consistency. *Eur J Oper Res* 2019; 273: 575–584.
12. Blakely F, Bozkaya B, Cao B, et al. Optimizing periodic maintenance operations for Schindler elevator corporation. *Interfaces* 2003; 33: 67–79.
13. Jang W, Lim HH, Crowe J, et al. The Missouri lottery optimizes its scheduling and routing to improve efficiency and balance. *Interfaces* 2006; 36: 302–313.
14. Gaur V and Fisher ML. A periodic inventory routing problem at a supermarket chain. *Oper Res* 2004; 52: 813–822.
15. Claassen GDH and Hendriks THB. An application of special ordered sets to a periodic milk collection problem. *Eur J Oper Res* 2007; 180: 754–769.
16. Mourgaya M and Vanderbeck F. Column generation based heuristic for tactical planning in multi-period vehicle routing. *Eur J Oper Res* 2007; 183: 1028–1041.
17. Francis PM, Smilowitz KR and Tzur M. The period vehicle routing problem and its extensions. The vehicle routing problem: latest advances and New challenges. In: Golden B, Raghavan S and Wasil E (eds) *Operations research/computer science interfaces series*. US: Springer, 2008, pp. 73–102.
18. Teixeira J, Antunes AP and de Sousa JP. Recyclable waste collection planning—a case study. *Eur J Oper Res* 2004; 158: 543–554.
19. Baptista S, Oliveira RC and Žuquete E. A period vehicle routing case study. *Eur J Oper Res* 2002; 139: 220–229.
20. Russell RA and Gribbin D. A multiphase approach to the period routing problem. *Networks* 1991; 21: 747–765.
21. Gaudioso M and Paletta G. A heuristic for the period vehicle routing problem. *Transp Sci* 1992; 26: 86–92.
22. Cordeau JF, Gendreau M and Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 1997; 30: 105–119.
23. Drummond LMA, Ochi LS and Vianna DS. An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Gener Comput Syst* 2001; 17: 379–386.
24. Hemmelmayr V, Doerner K and Hartl R. A variable neighborhood search heuristic for periodic routing problems. *Eur J Oper Res* 2009; 195: 791–802.
25. Tenahua A, Olivares-Benítez E, Sánchez-Partida D, et al. ILS metaheuristic to solve the periodic vehicle routing problem. *Int J Combin Opt Probl Inf* 2018; 9: 55–63.
26. Campbell AM and Wilson JH. Forty years of periodic vehicle routing. *Networks* 2014; 63: 2–15.
27. Mirjalili S and Lewis A. The whale optimization algorithm. *Adv Eng Soft* 2016; 95: 51–67.
28. Holland JH. Genetic algorithms. *Sci Am* 1992; 267: 66–72.
29. Kennedy J and Eberhart R. Particle swarm optimization. In: Proceedings of the 1995 IEEE international conference on neural networks, Perth, Australia, November 27–December 1 1995, pp. 1942–1948.
30. Jiang Y, Hao K, Cai X, et al. An improved reinforcement-immune algorithm for agricultural resource allocation optimization. *J Comput Sci* 2018; 27: 320–328.
31. Hsieh YC, Lee PJ and You PS. Immune based evolutionary algorithm for determining the optimal sequence of multiple disinfection operations. *Sci Iran E* 2019; 26: 959–974.
Appendixes

Appendix 1

Notation

\[ C = \{ 1, 2, \ldots, n \} \]

set of customers.

\[ C^k \]

set of nodes for customers with demand frequency \( k \), \( k \in F \). If \( k = 2 \), then the customer has to be served twice in \( T \) by using the type of 100100, 010010, or 001001. If \( k = 3 \), then the customer has to be served 3 times in \( T \) by using the type of 101010 or 010101. If \( k = 6 \), then the customer has to be served 6 times in \( T \) by using the type 111111.

\[ D \]

node of the depot.

\[ d_{ij} \]

shortest distance from node \( i \) to node \( j \), \( i, j \in C \cup \{ D \} \).

\[ F = \{ 2, 3, 6 \} \]

set of possible demand frequency for customers.

\[ h_t(w) \]

number of delivery for the vehicle in the time window \( w \) on day \( t \), \( w \in W, 1 \leq t \leq T \).

\[ n \]

total number of customers.

\[ N = 2 \times |C^2| + 3 \times |C^3| + 6 \times |C^6| \]

\[ P_t(w) = 0 \]

if \( R_t(w) \leq U_t(w) \), otherwise, \( P_t(w) = p(R_t(w) - U_t(w)) \) if \( R_t(w) > U_t(w) \), \( w \in W \) and \( 1 \leq t \leq T \). That is, if the vehicle can finish the scheduled delivery in the time window \( w \) on day \( t \), then the penalty is 0, otherwise the penalty is set as \( P_t(w) = p(R_t(w) - U_t(w)) \), where \( p \) is the penalty parameter. It is set to \( p = 100 \) in this paper.

\[ Q \]

capacity of the vehicle.

\[ q_i \]

demand of customer \( i \), \( i \in C \).

\[ R_t(w) = \sum_i \sum_j d_{ij}x_{ij}(w) \]

total traveling distance of the vehicle for all customers in the time window \( w \) on day \( t \), \( w \in W, 1 \leq t \leq T \).

\[ T \]

planning time horizon. It is set to \( T = 6 \) (days) in this paper.

\[ U_t(w) \]

upper limit of the traveling distance of the vehicle in the time window \( w \) on day \( t \), \( w \in W, 1 \leq t \leq T \).

\[ V_t(w) = V_{t_1}(w) \cup V_{t_2}(w) \cup \cdots \cup V_{t_g}(w) \cup \cdots \cup V_{t_h}(w) \]

set of the delivery sequence of customers for the vehicle in the time window \( w \) on day \( t \), where \( V_{tg}(w) \subseteq C \) is the \( g \)th tour sequence of the vehicle, \( 1 \leq g \leq h_t(w), w \in W, 1 \leq t \leq T \).

\[ W = \{ \text{a.m., p.m.} \} \]

set of the delivery time window for the vehicle.

\[ W_c = \{ \text{a.m., p.m., a.m./p.m.} \} \]

set of the possible time window for customers, \( w \in W_c \).

If \( w = \text{a.m./p.m.} \), then the delivery can be at a.m. or p.m. for the customer.
**Decision variables**

\[ x_{ij}(w) = 1 \] if customer \( i \) is served before customer \( j \) in the time window \( w \) on day \( t \), otherwise 0; \( i, j \in C, i \neq j, w \in W, 1 \leq t \leq T \).

\[ y_{it}(w) = 1 \] if customer \( i \) is served in the time window \( w \) on day \( t \), otherwise 0, \( i \in C, w \in W, 1 \leq t \leq T \).

**Appendix 2**

For the data of gas company (demand node and call time), see Table 8.

**Appendix 3**

For the data of the gas company (demand quantity), see Table 9.
Table 8. The data of gas company (demand node and call time).

| Call time      | Week 1 | Week 2 | Week 3 | Week 4 |
|----------------|--------|--------|--------|--------|
|                | D1     | D2     | D3     | D4     | D5     | D6     | D1     | D2     | D3     | D4     | D5     | D6     | D1     | D2     | D3     | D4     | D5     | D6     |
| 10:00–11:00    | 162    | 162    | 151    | 162    | 162    | 162    | 146    | 143    | 162    | 146    | 143    | 162    | 162    | 146    | 143    | 162    | 146    | 143    | 162    |
| 11:00–11:30    | 160    | 143    | 137    | 143    | 145    | 143    | 146    | 143    | 154    | 146    | 143    | 143    | 143    | 143    | 143    | 143    | 143    | 143    | 143    |
| 11:30–12:00    | 144    | 154    | 146    | 165    | 172    | 148    | 144    | 154    | 144    | 169    | 153    | 156    | 163    | 156    | 149    | 147    | 145    | 147    | 144    |
| 12:00–12:30    | 168    | 161    | 147    | 154    | 144    | 150    | 163    | 145    | 143    | 160    | 156    | 148    | 145    | 153    | 152    | 148    | 154    | 148    | 150    |
| 12:30–13:30    | 159    | 145    | 165    | 167    | 148    | 168    | 168    | 168    | 156    | 166    | 172    | 164    | 144    | 154    | 144    | 161    | 170    | 161    | 144    |
| 13:30–14:30    | 149    | 147    | 160    | 147    | 147    | 150    | 147    | 147    | 150    | 150    | 150    | 150    | 150    | 150    | 150    | 150    | 150    | 150    | 150    |
| 14:30–15:30    | 158    | 158    | 156    | 160    | 146    | 156    | 153    | 149    | 153    | 165    | 152    | 160    | 149    | 158    | 160    | 167    | 143    | 156    | 147    |
| 15:30–16:30    | 167    | 157    | 170    | 167    | 160    | 145    | 165    | 171    | 167    | 160    | 168    | 154    | 147    | 170    | 156    | 172    | 163    | 161    | 158    |
| 16:30–17:30    | 147    | 153    | 143    | 155    | 153    | 158    | 152    | 156    | 154    | 148    | 144    | 157    | 150    | 165    | 144    | 157    | 150    | 167    | 154    |
| 17:30–18:00    | 166    | 166    | 169    | 157    | 156    | 154    | 148    | 157    | 150    | 165    | 144    | 157    | 150    | 165    | 144    | 157    | 150    | 167    | 154    |
| 18:00–19:00    | 146    | 157    | 143    | 159    | 156    | 161    | 154    | 157    | 158    | 158    | 161    | 161    | 152    | 158    | 158    | 143    | 149    | 158    | 167    |
| 19:00–21:00    | 170    | 151    | 154    | 149    | 170    | 163    | 170    | 169    | 153    | 165    | 150    | 160    | 152    | 161    | 147    | 164    | 165    | 165    | 152    |
|                |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
Table 9. The data of the gas company (demand quantity).

| Demand node | Week 1 | Week 2 | Week 3 | Week 4 | Approx. frequency type (k) | Demand quantity |
|-------------|--------|--------|--------|--------|---------------------------|-----------------|
| 143         | 1      | 1      | 1      | 1      | 1                         | 6               |
| 144         | 2      | 2      | 2      | 2      | 2                         | 2               |
| 145         | 1      | 1      | 1      | 1      | 1                         | 2               |
| 146         | 1      | 1      | 2      | 1      | 2                         | 6               |
| 147         | 1      | 1      | 1      | 1      | 1                         | 6               |
| 148         | 3      | 2      | 1      | 1      | 2                         | 3               |
| 149         | 2      | 2      | 1      | 2      | 1                         | 2               |
| 150         | 1      | 1      | 1      | 1      | 1                         | 2               |
| 151         | 1      | 1      | 1      | 1      | 1                         | 3               |
| 152         | 1      | 1      | 1      | 1      | 1                         | 3               |
| 153         | 1      | 1      | 1      | 1      | 1                         | 3               |
| 154         | 1      | 1      | 1      | 1      | 1                         | 6               |
| 155         | 1      | 1      | 1      | 1      | 1                         | 2               |
| 156         | 1      | 1      | 1      | 1      | 1                         | 6               |
| 157         | 1      | 1      | 1      | 1      | 1                         | 3               |
| 158         | 1      | 2      | 1      | 1      | 1                         | 6               |
| 159         | 1      | 1      | 1      | 1      | 1                         | 3               |
| 160         | 2      | 2      | 1      | 1      | 1                         | 6               |
| 161         | 1      | 1      | 1      | 1      | 1                         | 2               |
| 162         | 1      | 1      | 1      | 1      | 1                         | 2               |
| 163         | 1      | 1      | 1      | 1      | 1                         | 3               |
| 164         | 1      | 1      | 1      | 1      | 1                         | 3               |
| 165         | 1      | 1      | 1      | 1      | 1                         | 6               |
| 166         | 1      | 1      | 1      | 1      | 1                         | 2               |
| 167         | 2      | 2      | 2      | 2      | 2                         | 6               |
| 168         | 1      | 2      | 1      | 1      | 1                         | 2               |
| 169         | 2      | 2      | 1      | 1      | 2                         | 2               |
| 170         | 1      | 1      | 2      | 1      | 1                         | 3               |
| 171         | 1      | 1      | 1      | 1      | 1                         | 2               |
| 172         | 1      | 1      | 2      | 2      | 1                         | 2               |