SOLVING A NEW MATHEMATICAL MODEL FOR A PERIODIC VEHICLE ROUTING PROBLEM BY NEIGHBORHOOD SEARCH METHOD

MARLAN¹, HERMAN MAWENGKANG²

¹Universitas Simalungun, Pematang Siantar, Indonesia
²Department of Mathematics, Universitas Sumatera Utara, Medan, Indonesia
E-mail: ²hmawengkang@yahoo.com

ABSTRACT

For a Periodic Vehicle Routing Problem (PVRP), this study offers a new mathematical model which optimizing vehicle travel expenses based on many assumptions. Periodic planning includes consideration of the following four issues that is a vehicle routing problem with time windows (VRPTW), a capacitated vehicle routing problem (CVRP), a vehicle routing problem with split service (VRPSS) and a vehicle routing problem with simultaneous pickup and delivery (VRPSPD). In large-scale issues, the computational complexity of this problem can be handled by any optimization program in a reasonable amount of time since it is based on a single model that we have created. In this paper a neighborhood search meta-heuristic is proposed.

Keywords: Periodic VRP, Logistics, Split Service, Meta-heuristic, Neighborhood Search

1. INTRODUCTION

The vehicle routing problem (VRP) is a general issue for specifying homogenous sets of routes and vehicles, wherein every vehicle begins at a garage and travels alongside a path to service a number of clients with identified geographic positions, and returns back to the garage when tour ends. The service might be anything from delivering things to picking up items to name a few. An essential part of the VRP is the depot, which stores and employed the vehicles to transport items to and from the depot, and the clients who receives the items. Reducing the overall transport route price while adhering to thorough going working time and vehicle capability limits is a fundamental goal of a VRP [1]. In spite of this, there may be various gaps among the fundamental VRP, and real-life implementation, for instance, depots quantity, client needs numerous pickups and deliveries, vehicles type with varied travel durations, capacity and expanses of travel, path restrictions and time frames of vehicles and so on. Variations, formulations, and solution procedures of VRPs were all examined in-depth by [2]–[4].

In VRPs, Capacitated vehicle routing problem (CVRP) is the most often studied. Looking at a route’s overall demand and VRP with time windows (VRPTW) in order to make sure that the vehicle can handle it all [5] and [6]. [7] developed Open Vehicle Routing Problem (OVRP) with a novel mathematical model. The model utilizes reasonable time windows by means of which distributors want to serve clients sooner than competitors to maximize sales. They developed a multi-objective particle swarm optimization (MOPSO) approach as well as a widely used multi-objective evolutionary method (NSGA-II) was used to compare their outcomes. Extending vehicle routing models has also been attempted for example several pickups and delivery positions VRP [8]–[10]. In addition, there is VRP with Simultaneous Pickup and Delivery (VRPSPD) wherein clients want both delivery and collection of items at the same time. There is a complete dynamic VRP model in [11], [12]. In the classical VRP, only one vehicle may service each node. Alternatively, more than one vehicle may service the client by numerous vehicles that pass through from that node. It is recognized as the VRP with the same split service (VRPSS) means that a service may be distributed across numerous vehicles [13].

Using VRP, one may come up with a schedule of services that is as cheap as possible [14]. A variation of VRP that develops routes is called PVRP. It was initially proposed as the Assignment Routing Problem in 1979, a mathematical model version of the Periodic Vehicle
Routing Problem (PVRP) [15]. Minimum numbers of visits are established as every day nodes are not visited. After that, research was refined to take into consideration number of days that individual node had been visited [16]. As a result of this, PVRP research developed exponentially. Numerous sorts of items have been studied using PVRP in the past. It demonstrates that these issues really occur on a daily basis. In PVRP study, a variety of materials were utilized as an insight such as distribution of vegetables [17], auto parts [18], collecting garbage [15, 19], beverage distribution [20], utility services [21], home health care (HCC) logistics [22], and many others. The majority of PVRP research has been focused on developing heuristic approaches for solving problems. Using the heuristic technique, you can come up with an acceptable solution to your issue, but you can’t be certain that you’ll obtain the best one [23]. Heuristics method may be useful when the number of customers serviced is enormous. Several researchers have been looking for a PVRP heuristic solution, like using variable neighborhood search [24], neighborhood search [25], particle swarm [26], hybrid genetic algorithm [27], hybridization of tabu search [22], hybrid metaheuristic algorithm [28], and large adaptive neighborhood [29]. Numerous advancements have been made in PVRP research. In terms of PVRP, there are three primary groups: multi-depot PVRP (MDPVRP), PVRP with time windows (PVRPTW), and PVRP with service choice (PVRP-SC) [30]. [21], [31] demonstrate the presence of multi-depot PVRP. [24], [25], [31]–[34] explain PVRP using time windows. PVRP-SC, on the other hand, refers to PVRP that allows number of visits to be used as judgment variable in discovering solutions, as in study [35]. It is claimed that PVRP-SC exhibits properties similar to those seen in the Inventory Routing Problem (IRP). PVRP-SC and IRP’s closeness to one another determines the visits frequency, arrangement of route, and deliveries in total[30]. The node’s schedule determines the PVRPSC attributes combination and the quantity of items sent to the node. Among the features of the IRP issue is quantity of deliveries at node is a decision variable varies from the number of visits [35]. FPVRP was used to develop this issue [35] by adding its flexible qualities to it [32]. The term “flexibility” refers to the ability to alter the frequency and number of visits. Flexibility was factored by [32] which gave a fresh viewpoint on modeling. Heuristic technique by [36] was one research that looked at flexibility. An FPVRP algorithm solution was found by [32] in two phases. Initial solutions were developed in the first phase, which was then followed by a tabu search process [37]. Additional advances in the two-tiered distribution channel with a flexible service time frame have been made [38]. Taking into consideration the consumer’s discount, a model was also designed which allows for more flexibility in delivery time [39].

The structure of the research article is maintained as follows. First, an overview of the topic and a review of relevant literature are the first items on the agenda. Section 2 defines the mathematical model, and Section 3 presents the suggested approach based on neighborhood search for resolving the presented issue. In section 4, calculations and results are presented. Lastly, Section 5 sums up the findings of this article and recommends prospective research avenues for further investigation.

2. MODEL FORMULATIONS

Following are some examples of how this model might be described. Let \( G = (V, A) \) be a chart where \( V = \{v_0, v_1, ..., v_n\} \) and \( A = \{(v_i, v_j)|v_i, v_j \in V, i \neq j\} \) area set of arcs and a set of nodes, in that order. \( A \) has two matrices, one for the expense of travel \( (e_{ij}) \) and one for the time it takes to get there \( (t_{ij}) \). Vertex \( v_0 \) is a depot, while the other vertices represent \( n \) clients. A set of permissible visitation times for each client is represented by \( H_i = \{S_{i1}, ..., S_{ih}\} \) and visitor \( S \)’s visitation schedule outlined by \( S = \{t_{i1}, ..., t_{iT}\} \). The customer’s demand on day \( d \) is indicated by the variable \( l_d \) (e.g., \( l_d = 0 \) shows services to on day \( d \) of the customer), and number of days sets are shown in the form of time period \( T \). Following are the notations used [40], [41].

2.1 Notations and variables

- \( n \) Clients total number
- \( m \) Vehicles total number
- \( t \) Days period
- \( c_{ij} \) Travel expense along arc \( (i, j) \)
- \( t_{ij} \) Travel period along arc \( (i, j) \)
- \( O_i \) Service period for client \( i \)
- \( p_i^d \) Pick up amount for client \( i \) in \( d \) – the period of day
- \( q_i^d \) Delivery amount of client \( i \) in \( d \) –the period of the day
- \( Q_k \) Capacity of vehicle \( k \)
- \( D_k \) Time frames for vehicle \( k \) to serve all clients
- \( t_{i}^{min} \) Lower limit of time frames for client \( i \)
- \( t_{i}^{max} \) Upper limit of time frames for client \( i \)
2.2 Decision variables

\[ v_{ij}^d = \begin{cases} 1 & \text{if schedule } S_i \text{ is selected for client } i \\ 0 & \text{otherwise} \end{cases} \]

\[ \theta_{jk}^d \quad \text{Quantity of pick-up orders of client } j \text{ vehicle } k \text{ served in } d \text{ day} \]

\[ \sigma_{jk}^d \quad \text{Quantity of deliveries orders of client } j \text{ vehicle } k \text{ served in } d \text{ day} \]

\[ \delta_{jk}^d \quad \text{Initial service period of client } j \text{ by vehicle } k \text{ in } d \text{ day} \]

s.t.

\[ \sum_{j=0}^{n} x_{ijk}^d = y_{ik}^d ; 1 \leq i \leq n, 1 \leq k \leq m, 1 \leq d \leq T \]  

(2)

\[ \sum_{i=0}^{n} x_{ikj}^d = y_{ik}^d ; 1 \leq j \leq n + 1, 1 \leq k \leq m, 1 \leq d \leq T \]  

(3)

\[ x_{ikj}^d \leq x_{ijk}^d Q_j^d ; 1 \leq i \leq n, 1 \leq k \leq m, 1 \leq d \leq T \]  

(4)

\[ \sum_{k=1}^{m} \theta_{jk}^d = p_j^d ; 1 \leq j \leq n, 1 \leq d \leq T \]  

(5)

\[ \sum_{k=1}^{m} \sigma_{jk}^d = q_j^d ; 1 \leq j \leq n, 1 \leq d \leq T \]  

(6)

\[ \delta_{n+1,k}^d - \delta_{nk}^d \leq D_k^d ; 1 \leq k \leq n, 1 \leq d \leq T \]  

(7)

2.3 Mathematical model

\[ \min Z = \sum_{d=1}^{T} \sum_{i=0}^{n} \sum_{j=0}^{n+1} c_{ij} x_{ijk}^d \]  

(1)

\[ \sum_{i=0}^{n} v_{ij}^d = \begin{cases} 1 & \text{if day } d \text{ is in schedule } S_i \\ 0 & \text{otherwise} \end{cases} \]

\[ \sum_{i=0}^{n} v_{ij}^d \times a_{ij}^d = \begin{cases} 1 \Rightarrow \sum_{i=1}^{n} y_{ij}^d \geq 1 ; 1 \leq i \leq n, 1 \leq d \leq T \\
0 \Rightarrow \sum_{i=1}^{n} y_{ij}^d = 0 \end{cases} \]  

(12)

\[ \sum_{i=0}^{n} \sum_{j=0}^{n+1} x_{ijk}^d \leq |P| - 1 ; 1 \leq d \leq T \]  

(13)

\[ x_{ijk}^d, y_{ik}^d, v_{ij}^d \in \{0, 1\} ; \forall (i, j, k, d) \]  

(14)

\[ z_{ijk}^d, \theta_{jk}^d, \sigma_{jk}^d \in \{0, 1, 2, \ldots\} ; \forall (i, j, k, d) \]  

(15)

Keeping the cost of the route to a minimum is the goal of this function. Constraints (1) and (2) make certain that each vehicle that comes at a client’s location needs to depart from that location as soon as feasible (1), this requires that each vehicle allocated to a client must service every single day (2). Constraint (3) is designed to avert capacity of the vehicle from exceeding limit as vehicle \( k \) travels the arc \((i, j)\). At a minimum, the relevant load \( (x_{ijk}^d) \) should not surpass the capacity of vehicle \( (Q_j^d) \). In order to meet the daily pickup and delivery needs of each client. Constraints (4) and (5) mandate the use of a vehicle transit system. The maximum time a vehicle may be in service is outlined in Constraint (6). After a client has been served, the vehicle’s load is balanced by constraint (7). Each vehicle must arrive at address of the customer within the time limit established by the node of constraint (8). Constraint (9) confirms if \( x_{ijk}^d = 1 \), customer’s location \( j \) must have an arrival time greater than the total of the customer’s location \( S_i \) arrival times, customer i time of service and arc time of travel \((i, j)\). For each client, only one visitation schedule is available constraint (10). For each client, Constraint (11) expresses in the number of days in their selected schedule, which may be supplied by many vehicles; or else, no visit will be made on a certain day if it does not fall within the predetermined timetable. The sub-tour elimination constraint is represented by constraint (12). Last but not least, constraint (13) maintains the integrality of the variables in the model.

3. THE SOLUTION BASIC APPROACH

Look at an example mixed integer linear programming (MILP) issue that has the succeeding structure.

Minimize \( P = c^T x \)  

Subject to \( Ax \leq b \)  

\( x \geq 0 \)  

\( x_j \) number for approximately \( j \in J \) (\( J \) is index set)
the basic feasible vector \((x_B)_k\) component in terms of MILP solution as uninterrupted shown below in equation 16

\[
(x_B)_k = \beta_k - \alpha_{kj} (x_N)_j - \cdots - \alpha_{kj} (x_N)_j - \cdots - \alpha_{kj} (x_N)_j \rightarrow \alpha_{kj} (x_N)_j \geq 0 (16)
\]

This statement can be seen in the Simplex method’s final tableau. For example, if \((x_B)_k\) would be an integer variable and we suppose \(k\) is not, the fractional and integer elements of \(k\) are provided by

\[
\beta_k = [\beta_k] + f_k, \quad 0 \leq f_k \leq 1 \quad (17)
\]

Supposing bump \((x_B)_k\) up to the closest integer, say \([\beta] + 1\). We might raise a non-basic variable on the basis of concept of suboptimal elucidations, such as \((x_N)_j\), exceeding the zero limit, given that \(\alpha_{kj}\) is a negative component of the vector \(\alpha_x\). Let \(\Delta_j\) be total movement of the non-variable \((x_N)_j\), and therefore the scalar \((x_B)_k\) numerical value is an integer \(\Delta_j\) can then be stated using Eqn. (16) as follows.

\[
\Delta_j = \frac{1-f_k}{-\alpha_{kj}} \quad (18)
\]

Non-basics remain at a constant 0. As may be observed, by replacing (18) into (16) for \((x_N)_j\), when we take into consideration (17)’s division of the number \(x\), then we get

\[
(x_B)_k = [\beta] + 1
\]

Therefore, now \((x_B)_k\) becomes an integer.

The importance of a non-basic variable in integerizing the equivalent basic variable has now been established. As a consequence, the following result is required to prove that the integerizing procedure must operate with a non-integer variable.

**Theorem.**

If the MILP issue (1)-(4) has an optimum answer, then non-basic variables must be included in the solution. \((x_N)_j, j = 1, \ldots, n\), should be non-integer variables.

**Proof.**

Using slack variables as a continuous in solving the issue (except in the case of equality constraints, which are non-integer). Assuming \(x_B\) is a basic variables of vector including total slack variables, so variables becomes integer values as they are included in the non-basic vector \(x_N\).

As the scalar’s \((x_N)_j\)’ numerical value increases to \(\Delta_j\), so will the additional components \((x_B)\) of vector \(x_B\). Thus, if vector \(\alpha_j^*\), i.e., \(\alpha_j\) for \(i \neq k\), some elements shows positive result. Thus, the \(x_B\) element starts decreasing and may potentially reach 0 at some point. Due to the non-negativity condition, however, vector \(x\) no components may becomes less than zero. Because of this, a method named the minimal proportion test is required to determine the non-basic’s \((x_N)_j\)’ maximum movement while still keeping all of \(x\) viable. Two aspects must be included in this ratio test.

1. Firstly, a basic variable \((x_B)_{i \neq k}\) reduces to 0 (lower bound).
2. The basic variable, \((x_B)_k\) rises to an integer.

If we were to apply these both aspects to each other, we’d do the following:

\[
\theta_1 = \min \left\{ \frac{\beta_j}{\alpha_{kj}} \right\} \quad (19)
\]

\[
\theta_2 = \Delta_j \quad (20)
\]

Amount of non-basic \((x_N)_j\)’ release that allows vector \(x\) to remain viable despite its zero-bound, depends on the \(\theta^*\)-ratio test as can be seen below

\[
\theta^* = \min \left( \theta_1, \theta_2 \right) \quad (21)
\]

evidently, if \(\theta^* = \theta_1\), one of the fundamental variables \((x_B)_{i \neq k}\) will reach its lower limit prior to the integer value for \((x_B)_k\). If \(\theta^* = \theta_2\), feasibility is preserved since the fundamental variable \((x_B)_k\) will have an integer value. In the same way, we may decrease the numerical value of the fundamental variable \((x_B)_k\) to its nearest integer \([\beta_k]\). Any positive \(\alpha_j^*\)-value in this situation corresponds to the movement amount caused by the \((x_N)_j\)’ non-basic variable and hence

\[
\Delta_j = \frac{f_k}{\alpha_{kj}} \quad (22)
\]

Ratio test \(\theta^*\) is still required to keep things feasible. \(\Delta_j\), reflect the particular non basic variable, as represented in equation (18) and (22). The vector \(\alpha_j^*\)’s corresponding element are the sole factor that researcher must consider during their operations. A vector \(\alpha_j\) may be represented as

\[
\alpha_j = B^{-1} a_j, \quad j = 1, \ldots, n - m \quad (23)
\]

As a result, to acquire a certain vector \(\alpha_j\) element one must first identify the associated matrix \([B]^{-1}\) columns. Let’s say that we want to find out the \(\alpha_{kj}^*\) value allowing \(v_k^T\) become the \(k\)-th column vector of \([B]^{-1}\), we will get

\[
v_k^T = \alpha_{kj}^* B^{-1} \quad (24)
\]

After that, we can calculate \(\alpha_{kj}^*\)’s numerical value from

\[
\alpha_{kj}^* = v_k^T \cdot a_j \quad (25)
\]
Eqns. (24) and (25) are stated to as pricing operation in Linear Programming (LP). $d_j$ represents the vector of decreased prices. It is utilized to assess the decrease in the objective function value produced by a non-basic variable releasing from its bounds. When choosing which non-basic to release in the integerization procedure, special consideration must be taken to the vector $d_j$, so that degradation is diminished. A lower limit on any integer-equivalent solution may be found using the minimal continuous solution. The amount of movement in a given non-basic variable as given in Eqn. (18) or (22), is, nevertheless, influenced by the associated $\alpha$-vector element. As a result, releasing a non-basic variable $(x_N)_j$ results in objective function reduced value. When determining how to integerize $x$, the ratio is:

$$\frac{d_k}{\alpha_{kj}}$$

(26)

Where $[\alpha]$ implies the definite value of scalar $\alpha$.

To keep the best continuous solution as close to zero as possible, we apply the following technique to determine which non-basic variable raises from its zero limit, that is,

$$\min_j \left| \frac{d_k}{\alpha_{kj}} \right| , j = 1, \ldots, n - m$$

(27)

Writing constraints for non-basic $(N)$, basic $(B)$, and superbasic $(S)$, variables may be done using a “active constraint” technique and the splitting of the constraints equivalent to these three variables.

$$
\begin{bmatrix}
B & S & N \\
I & & \\
& & \\
\end{bmatrix}
\begin{bmatrix}
x_b \\
x_S \\
x_N \\
\end{bmatrix}
= 
\begin{bmatrix}
b \\
\ \\
\end{bmatrix}
$$

(28)

or

$$Bx_b + Sx_S + Nx_N = b$$

(29)

It is supposed that matrix $B$ would be non-singular and square matrix but we obtained as follows.

$$x_B = \beta - Wx_S - \alpha x_N$$

(31)

Where:

$$\beta = B^{-1}b$$

(32)

$$W = B^{-1}S$$

(33)

$$\alpha = B^{-1}N$$

(34)

Non-basic variables are being kept to their limit in expression (30). Eqnn. (31), which uses the integerizing technique mentioned in the preceding section and suited for MILP problems, makes it clear that this strategy may be put into action. A degenerate solution would allow us to liberate a non-basic variable from the constraints of Eqnn. (30) and swap it for a basic variable of the same type after integerization.

The basic variable, $(x_B)_k$, is being integerized right now, and as a result, the non-basic variable, $(c_N)_j$ , is being liberating from its zero-bound. Assume that $(x_N)_j$’s maximum movement fulfills $\theta^* = \Delta_j$ .

In order to take use of the method of modifying the basis, $(x_B)_k$ must be integer valued, we transfer $(x_N)_j$, keen on $B$ (to substitute $(x_B)_k$) and integer-valued $(x_B)_k$ into $S$ to ensure that the integer solution remains intact. Now that a fundamental variable has reached its limit, we have a degenerate solution. With a fresh set of integers, the process of integerizing continues $[B, S]$. As a result, total number of integral variables becomes superbasic.

4. THE ALGORITHM

The following approach may be used to find a suboptimal but integer-feasible solution from an optimum continuous solution once the relaxed issue has been solved. Let

$x = [x] + f, \quad 0 \leq f \leq 1$

be the (continuous) solution of the tranquil issue, $[x]$ is the integer component of non-integer variable $x$ and $f$ is the fractional component.

Step 1. Obtaining the smallest integer feasibility of row $i^*$, $\delta_r = \min \{f_i, 1 - f_i\}$

Step 2. Compute

$$v^*_r = c^*_r B^{-1}$$

this operation is named as pricing operation.

Step 3. Compute

$$\sigma_j = v^*_r a_j$$

With $j$ match up to $\min_j \left\{ \frac{d_j}{\sigma_j} \right\}$

I. For non-basic j at lower limit

If $\sigma_j < 0$ and $\delta_r = f_i$ determine

$$\Delta = \frac{(1 - \delta_r)}{-\sigma_j}$$

If $\sigma_j > 0$ and $\delta_r = 1 - f_i$ determine

$$\Delta = \frac{(1 - \delta_r)}{\sigma_j}$$
If $\sigma_{ij} < 0$ and $\delta_{r} = 1 - f_{i}$ determine
\[ \Delta = \frac{\delta_{r}}{-\sigma_{ij}} \]
If $\sigma_{ij} > 0$ and $\delta_{r} = f_{i}$ determine
\[ \Delta = \frac{\delta_{r}}{\sigma_{ij}} \]

II. For non-basic $j$ at upper limit
If $\sigma_{ij} < 0$ and $\delta_{r} = 1 - f_{i}$ determine
\[ \Delta = \frac{(1 - \delta_{r})}{-\sigma_{ij}} \]
If $\sigma_{ij} > 0$ and $\delta_{r} = f_{i}$ determine
\[ \Delta = \frac{(1 - \delta_{r})}{\sigma_{ij}} \]
If $\sigma_{ij} > 0$ and $\delta_{r} = 1 - f_{i}$ determine
\[ \Delta = \frac{\delta_{r}}{-\sigma_{ij}} \]
If $\sigma_{ij} < 0$ and $\delta_{r} = f_{i}$ determine
\[ \Delta = \frac{\delta_{r}}{-\sigma_{ij}} \]

Else, go to available superbasic $j$ or non-integer non-basic variable. Finally column $j^*$ is elevated from LB or reduced from UB. If this never happen, then proceed to the next $i^*$.

Step 4. Compute
\[ \alpha_{j^*} = B^{-1} \alpha_{j^*} \]
i.e., solve $B \alpha_{j^*} = \alpha_{j^*}$ for $\alpha_{j^*}$.

Step 5. Ratio test: Because non-basic $j^*$ has been released from its limits, there are three possible values for the basic variables.
If $j^*$ lower limit
Let
\[ A = \min_{i \in \neq j^*} \left\{ \frac{x_{Bi} - l_i}{\alpha_{j^*}} \right\} \]
\[ B = \min_{i \in \neq j^*} \left\{ \frac{u_i - x_{Bi}}{-\alpha_{j^*}} \right\} \]
\[ C = \Delta \]
The $j^*$ maximum movement rely on:
\[ \theta^* = \min(A, B, C) \]
If $j^*$ upper limit
Suppose
\[ A' = \min_{i \in \neq j^*} \left\{ \frac{x_{Bi} - l_i}{\alpha_{j^*}} \right\} \]
\[ B' = \min_{i \in \neq j^*} \left\{ \frac{u_i - x_{Bi}}{-\alpha_{j^*}} \right\} \]
\[ C' = \Delta \]
The $j^*$ maximum movement rely on:
\[ \theta^* = \min(A', B', C') \]

Step 6. Switching basis for all probabilities
1. If $A$ or $A'$
   - $x_{Bi}$ benefits non-basic at lower limit $l_i$.
   - $x_{j^*}$ benefits basic (substitutes $x_{Bi}$)
   - $x_{j^*}$ remains basic non-integer
2. If $B$ or $B'$
   - $x_{Bi}$ becomes non-basic at upper limit $u_i$.
   - $x_{j^*}$ becomes basic (replaces $x_{Bi}$)
   - $x_{j^*}$ remains basic non-integer
3. If $C$ or $C'$
   - $x_{j^*}$ benefits basic (replaces $x_{i^*}$)
   - $x_{j^*}$ becomes super-basic at integer-valued

repeat from step 1.

5. Computational Result

Exit -- Optimal Solution Found.

<table>
<thead>
<tr>
<th>No. of Iterations</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective Value</td>
<td>4.5000000000000E+02</td>
</tr>
<tr>
<td>Norm of X</td>
<td>8.250E+02</td>
</tr>
<tr>
<td>Norm of Pi</td>
<td>1.735E+02</td>
</tr>
<tr>
<td>Problem Name</td>
<td>P</td>
</tr>
<tr>
<td>Objective Value</td>
<td>4.5000000000000E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Optimal Soln</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration</td>
<td>67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 1: The values of variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X011 0.00000</td>
</tr>
<tr>
<td>2 X021 0.00000</td>
</tr>
<tr>
<td>3 X031 0.00000</td>
</tr>
<tr>
<td>4 X041 1.00000</td>
</tr>
<tr>
<td>5 X012 0.00000</td>
</tr>
<tr>
<td>6 X022 0.00000</td>
</tr>
<tr>
<td>7 X032 1.00000</td>
</tr>
<tr>
<td>8 X042 0.00000</td>
</tr>
<tr>
<td>9 X101 1.00000</td>
</tr>
</tbody>
</table>
6. CONCLUSION

The prior models failed to consider the true complexity of many real-world routing issues. To suit a variety of practical needs, we've developed a PVRP, or periodic vehicle routing problem in this research, incorporating the PVRP’s well-known models. Furthermore, it incorporates plenty of formerly unconsidered upgrades from prior models. This paper has presented a split service of CPVRP that includes the ability to divide a customer’s transportation needs across several vehicles. In a transportation system, this issue might arise when a large number of vehicles must pass through a node or client. Another possibility is that the order in certain nodes exceeds the fleet’s total capacity. For the purpose of this research, we have sought to optimize the fleet’s capacity utilization. As a result, numerous vehicles might meet the needs of certain clients. The provided approach is capable of finding the most cost-effective routes for a fleet. It is clear that the suggested PVRP model can be solved using the PSO method, as shown by the simulation results. The PSO parameters and programming implementation may be improved, though, since it was not the greatest. A better solution and a shorter calculation time are now possible thanks to these additional efforts. Additional studies should be done to expand the approach to more complex real-world issues.

REFERENCES:


