Jaya Algorithm + Savings + 2-Opt Heuristic for Multi-Objective Capacitated Vehicle Routing Problem with Time Constraints & Heterogeneous Fleet of Vehicles

Prof. R. J. Dhake*
Associate Professor, Industrial & Production Engineering Department, Vishwakarma Institute of Technology, Pune, India.

Dr. N.R. Rajhans
Professor, Production Engineering & Industrial Management Department, College of Engineering, Pune, India.

Nagesh Bhole
S.Y. B. Tech. Computer Engineering Department, Vishwakarma Institute of Technology, Pune, India.
*(Corresponding Author)

Abstract
Vehicle Routing Problem (VRP) plays a vital role in distribution and logistics. The current work focuses on optimum routing of a fleet of pick-up vehicles that collect parts and components from a large number of suppliers (of consumer durables manufacturer) located in the three municipal corporations’ viz., Bhuwai Municipal Corporation, Navi Mumbai Municipal Corporation & Thane Municipal Corporation and a collection center. The objective is to minimize the total distance travelled, which is directly proportional to the transportation costs and minimizing the number of pick-up vehicles. Dynamic nature of the problem restricts use of exact solution algorithms. The solution is found using Jaya Algorithm (Meta Heuristic Approach - a population-based new efficient optimization method that generates a population of solutions to proceed to the global solution) & Savings Algorithm with 2-opt improvement heuristic.

Keywords: Vehicle Routing Problem, Meta-heuristics, Jaya Algorithm

INTRODUCTION
The Vehicle Routing Problem (VRP), a generalized case of Traveling Salesman Problem (TSP) introduced by Dantzig and Ramser (Dantzig, 1959) holds a central place in logistics management (both inbound and outbound logistics) and is one of the most widely studied problems in combinatorial optimization. In the initial period solutions generated were using classical heuristics viz. sweep, strip, nearest neighbor, minimal spanning tree, savings algorithm, etc. In the past two decades however, the much of the research shows the use of meta-heuristics using mainly two search methods: local search and population search (Cordeau, 2002).

Vehicle routing problems have been the subject of intensive research for more than 50 years, due to their great scientific interest as difficult combinatorial optimization problems and their importance in many application fields, including transportation, logistics, communications, manufacturing, military and relief systems, and so on (Vidal, 2013). Vidal T. et al, 2013 classified the solution methods for VRP into four categories: constructive heuristics, improvement heuristics, meta-heuristics, hybrid methods, and parallel and cooperative meta-heuristics.

The term “meta-heuristic” was first coined by Glover (1986) to designate a broad class of heuristic methods that continue the search beyond the first encountered local optimum. A somewhat crude but telling definition characterizes meta-heuristics as heuristics guiding other heuristics. According to Laporte G, et al, 1986, several meta-heuristics have been proposed for the VRP. These are general solution procedures that explore the solution space to identify good solutions and often embed some of the standard route construction and improvement heuristics. In a major departure from classical approaches, meta-heuristics allow deteriorating and even infeasible intermediary solutions in the course of the search process. The best known meta-heuristics developed for the VRP typically identify better local optima than earlier heuristics, but they also tend to be more time consuming (Laporte, 1999).

The case refers to inbound logistics network of consumer durable manufacturer. In this 2-echelon (2 stages) network, pick up operations are done in two stages. The material is first collected from 28 suppliers (230 components) located in three municipal corporations viz., BMC, NMMC & TMC in the Mumbai region through pick-up / mini-door vehicles (light commercial vehicles) and consolidated at the collection centre. This is then cross docked in a single Tata 1610/1613 truck (heavy commercial vehicle) and dispatched to the manufacturer located 250 km away on a daily basis. A private transport contractor charges the manufacturer at pre-determined rates mutually decided through negotiations.

LITERATURE SURVEY
A large variety of algorithms (solution methods) are available to solve VRP. They primarily fall into three categories viz., exact solution algorithms, heuristics and meta-heuristics. The past two decades have found a growing use of meta-heuristic techniques which are mostly nature-inspired optimization algorithms. The metaheuristics explore the global optima without getting stuck in local optima thereby substantially improving the quality of solutions. Several metaheuristics
(Ant Colony Algorithm, Genetic Algorithm, Tabu Search, Simulated Annealing, Particle Swarm Optimization, Greedy Random Adaptive Search Methods, Honey Bee Mating Optimization, etc.) have been adapted for solving a large variant of VRP (VRP, Capacitated VRP, VRP with Time Windows, Heterogeneous VRP, Multi-Deport VRP, etc.)

**Jaya Algorithm**

Jaya Algorithm proposed by R.V. Rao (R.V. Rao, 2016) is based on the concept that the solution obtained for a given problem should move towards the best solution and should avoid the worst solution. Since, then it has been used in different applications (Zhang, et al 2016; R.V. Rao, et al 2016). Fig.1 shows the flowchart of the algorithm. The algorithm always tries to get closer to success (i.e. reaching the best solution) and tries to avoid failure (i.e. moving away from the worst solution).

![Flowchart of Jaya Algorithm](image-url)

**Jaya Clustering**

Clustering procedure using Jaya Algorithm is explained below:

1. Initialize each candidate to contain $N$, randomly selected cluster centroids.
2. For $t = 1$ to $t_{\text{max}}$ do
   a. For each candidate $i$ do
   b. For each data vector $z_e$
      i. Calculate the distance $d(z_e, m_u)$ to all cluster centroids $C_u$
   c. Update the candidate modification
   d. Update the cluster centroids using equations (1) and (3)

   where $t_{\text{max}}$ is the maximum number of iterations

**Clarke and Wright Savings Heuristic**
The Clarke and Wright savings heuristic is the most popular heuristic, which has withstood the test of time on account of its speed, simplicity and reasonably good accuracy (Clarke, 1964). The solution quality is improved using 2-Opt method.

1. Problem Description

The firm under consideration is a consumer durable manufacturer which collects parts and components (more than 560 in variety) from its suppliers (nodes) located in three municipal corporations’ viz., Bruhan Mumbai Municipal Corporation, Navi Mumbai Municipal Corporation & Thane Municipal Corporation. The pickup operations are outsourced to a third-party transporter and are based on past experience. The vehicles need to ply through the densely populated areas and need to adhere to time windows and capacity constraints. These supplies are then brought to a collection center (depot) from where these are subsequently shipped to the manufacturing plant through a heavy commercial vehicle on a daily basis. In the present practice, four routes are identified for collecting material from 32 suppliers (nodes) to collection center (depot) through light commercial vehicles of varying capacities. The schedule, frequency of collection and other details of pickup operations, including the transportation costs is fixed and given in table 1.

The present method of pickup operations highlights the following problems:

- No consideration given to volume and weight of material transported; vehicle capacities and loading time required at suppliers. The manufacturer is charged at a flat rate.
- Need for additional trips than scheduled on account of excess volumes collected from suppliers which are more than vehicle capacity. This often leads to additional trips and transportation costs. This approximately amounts to 20% of the estimated costs.
- The transporter charges the manufacturer at a flat rate, which leads to excessive transportation costs. Routes are not optimized considering the volume of material collected, distance between suppliers, vehicle capacities.
- Need to optimize the pickup operations ensuring FTL (full truck load), thereby minimizing number of vehicles required and transportation costs.

PROPOSED METHODOLOGY

Data Collection

The first step is to collect the relevant information related to the following parameters:

- Location of Collection Centre & Suppliers: Distances in km between suppliers and collection centre
- Supplier wise Component Details: Item Code, Volume per unit.
- Daily Requirement for Each Component: As Per Production Plan.
- Supplier wise Collection Volumes: (Based on Collection Frequency)
- Loading Time per Supplier: Total volume collected per trip and loading time required at each supplier.
- Type of Vehicles Available: Number of vehicles of each type with capacities in volume.

<table>
<thead>
<tr>
<th>Route</th>
<th>Region (Location of Suppliers) Collection Schedule (Frequency / week)</th>
<th>No. of Suppliers</th>
<th>Cost Rate Per Trip (₹)</th>
<th>Avg. No. of Trips Per Month</th>
<th>Total Cost Per Route (₹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Central Suburban Areas</td>
<td>16</td>
<td>1850</td>
<td>35</td>
<td>64,750</td>
</tr>
<tr>
<td>2</td>
<td>Western Suburban Areas</td>
<td>9</td>
<td>1525</td>
<td>32</td>
<td>48,800</td>
</tr>
<tr>
<td>3</td>
<td>Greater Mumbai</td>
<td>3</td>
<td>1525</td>
<td>31</td>
<td>47,275</td>
</tr>
<tr>
<td>All Routes</td>
<td>Annual Transportation Costs (Including Extra Trips ) (₹)</td>
<td></td>
<td></td>
<td></td>
<td>1,60,825</td>
</tr>
</tbody>
</table>

Table 2: Coordinates, Demand (Cubic Feet) and Loading Time(min)
<table>
<thead>
<tr>
<th>Location</th>
<th>X Coordinates</th>
<th>Y Coordinates</th>
<th>Demand</th>
<th>Loading Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location 1</td>
<td>17.6</td>
<td>18.5</td>
<td>28.88861</td>
<td>60</td>
</tr>
<tr>
<td>Location 2</td>
<td>18</td>
<td>26</td>
<td>44.53981</td>
<td>60</td>
</tr>
<tr>
<td>Location 3</td>
<td>17</td>
<td>25</td>
<td>15.60541</td>
<td>60</td>
</tr>
<tr>
<td>Location 4</td>
<td>16.3</td>
<td>23</td>
<td>67.89277</td>
<td>30</td>
</tr>
<tr>
<td>Location 5</td>
<td>16.4</td>
<td>23</td>
<td>7.66687</td>
<td>30</td>
</tr>
<tr>
<td>Location 6</td>
<td>17</td>
<td>22.5</td>
<td>54.85593</td>
<td>30</td>
</tr>
<tr>
<td>Location 7</td>
<td>16.8</td>
<td>21.8</td>
<td>38.24583</td>
<td>30</td>
</tr>
<tr>
<td>Location 8</td>
<td>16.8</td>
<td>20.5</td>
<td>16.33708</td>
<td>30</td>
</tr>
<tr>
<td>Location 9</td>
<td>15.6</td>
<td>20.6</td>
<td>37.28839</td>
<td>30</td>
</tr>
<tr>
<td>Location 10</td>
<td>15.6</td>
<td>19</td>
<td>5.3888889</td>
<td>50</td>
</tr>
<tr>
<td>Location 11</td>
<td>15.6</td>
<td>19.1</td>
<td>181.05771</td>
<td>60</td>
</tr>
<tr>
<td>Location 12</td>
<td>14.8</td>
<td>19.2</td>
<td>437.22493</td>
<td>120</td>
</tr>
<tr>
<td>Location 13</td>
<td>14.8</td>
<td>18.2</td>
<td>309.26028</td>
<td>120</td>
</tr>
<tr>
<td>Location 14</td>
<td>15</td>
<td>18.4</td>
<td>8.7920833</td>
<td>40</td>
</tr>
<tr>
<td>Location 15</td>
<td>13.6</td>
<td>16.2</td>
<td>4.903814</td>
<td>40</td>
</tr>
<tr>
<td>Location 16</td>
<td>13</td>
<td>14</td>
<td>31.28125</td>
<td>20</td>
</tr>
<tr>
<td>Location 17</td>
<td>12.8</td>
<td>18.8</td>
<td>163.63236</td>
<td>40</td>
</tr>
<tr>
<td>Location 18</td>
<td>10.2</td>
<td>17.8</td>
<td>114.31632</td>
<td>30</td>
</tr>
<tr>
<td>Location 19</td>
<td>8.6</td>
<td>20.2</td>
<td>47.560556</td>
<td>40</td>
</tr>
<tr>
<td>Location 20</td>
<td>9</td>
<td>24.6</td>
<td>2004.5285</td>
<td>80</td>
</tr>
<tr>
<td>Location 21</td>
<td>8.6</td>
<td>24.5</td>
<td>160.06986</td>
<td>80</td>
</tr>
<tr>
<td>Location 22</td>
<td>8.3</td>
<td>24.2</td>
<td>5.7692361</td>
<td>20</td>
</tr>
<tr>
<td>Location 23</td>
<td>8</td>
<td>23.9</td>
<td>2.4343056</td>
<td>20</td>
</tr>
<tr>
<td>Location 24</td>
<td>7.5</td>
<td>23</td>
<td>1.7846611</td>
<td>20</td>
</tr>
<tr>
<td>Location 25</td>
<td>9</td>
<td>15</td>
<td>13.846181</td>
<td>20</td>
</tr>
<tr>
<td>Location 26</td>
<td>7.4</td>
<td>5.5</td>
<td>16.395833</td>
<td>60</td>
</tr>
<tr>
<td>Location 27</td>
<td>6.8</td>
<td>4.8</td>
<td>10.069093</td>
<td>60</td>
</tr>
<tr>
<td>Location 28</td>
<td>6.4</td>
<td>1.4</td>
<td>0.2316667</td>
<td>120</td>
</tr>
</tbody>
</table>

Table 3: Distance Matrix

| Table 4: Types of Vehicles Available |
Problem Formulation
A fleet of vehicles, based at a single collection centre, collects a number of consignments from 28 suppliers located in the 3 municipal corporation limits. All orders are consolidated to full truckloads and are subject to time window constraints. The underlying assumptions of our model are:
- The problem is static; all orders are known a priori.
- Transportation cost is directly proportional to the distance travelled.
- Backhauls are not permitted.
- All orders are consolidated to full truckloads and don’t exceed the vehicle capacity.
- Time windows for each order have to be adhered strictly.
- A tour must not exceed a given time-span. This assumption models legal restrictions on the maximum driving time for truck drivers. It could also be viewed as a restriction, which ensures that maintenance intervals for vehicles are respected.
- Each vehicle has to start and return to the collection centre after each tour.
- Since all components have high volume-weight ratio, the constraint for their transportation is volume rather than weight. The requirement of these components remains fixed over one month period and is specified by manufacturer through its firm production plan.
- Average Vehicle Speed: 20 kmph (Suppliers to Collection Centre)
- Working Hours At Suppliers: Between 8.00 am to 6.20 pm

Fisher and Jaikumar (1978, 1981) developed a three-index vehicle flow formulation for VRPs with capacity restrictions, time windows and no stopping times. Such formulations use variables to represent the passing of a vehicle on an arc or edge (i, j). In three-index formulations, variables \( x_{ijk} \) indicate whether (i, j) is traversed by vehicle k or not. In two-index formulations, variables \( x_{ij} \) do not specify which vehicle is used on (i, j). Define binary variables \( x_{ijk} \) (i\(\neq\)j), equal to 1 if and only if in the optimal solution, arc (i, j) is traversed by vehicle k. Also define binary variables \( y_k \) equal to 1 if and only if vertex i is served by vehicle k. The time windows are two-sided, meaning that a customer must be serviced at or after its earliest time and before its latest time. If a vehicle reaches a customer before the earliest time it results in idle or waiting time. A vehicle that reaches a customer after the latest time is tardy. A service time (loading / unloading) is also associated with servicing each customer. The route cost of a vehicle is the total of the traveling time (proportional to the distance) and service time taken to visit a set of customers.

### Decision Variables

VRP is a combinatorial problem whose ground set is the edges of a graph \( G = (V, E) \). The notation used for this problem is as follows:

- \( G = (V, E) \): the graph representing the vehicle routing network with vertices \( V = \{v_0, v_1, v_2, \ldots, v_n\} \) and
- \( E = \{(v_i, v_j) \mid v_i, v_j \in V, i < j \} \) is an arc set
- \( V = \{v_0, v_1, v_2, \ldots, v_n\} \) vertex set where \( v_0 \) represents depot location and \( v_1, v_2, \ldots, v_n \) represent customer locations
- \( n \) number of nodes corresponding to customer locations for each instance (trip or route)
- \( o \) depot location
- \( k = k_A + k_B \), number of pick-up vehicles of type A and B respectively
- \( m \) total number of trips corresponding to number of vehicles used
- \( d_{ij} \) distance (proportional to travel time) between vertices \( v_i \) and \( v_j \) (non-negative) \((i \neq j)\)
- \( t_{ij} \) travel time (proportional to distance) between vertices \( v_i \) and \( v_j \) (non-negative) \((i \neq j)\)
- \( e_i \) earliest arrival time at customer \( i \) \((i = 1, \ldots, n)\)
- \( l_i \) latest arrival time at customer \( i \) \((i = 1, \ldots, n)\)
- \( t_i \) total travel time to reach customer \( i \) \((i = 1, \ldots, n)\)
- \( s_i \) service time for customer \( i \) \((i = 1, \ldots, n)\)
- \( T \) maximum travel time permitted for vehicle \((T = 620)\)
- \( T_k \) total travel time (including service time) for a vehicle route \( k \) \((k = 1, \ldots, m)\)
- \( E_i \) earliest time allowed for pickup to customer \( i \)
- \( L_i \) latest time allowed for pickup to customer \( i \).
\( T \) = maximum travel time permitted for a vehicle

\( q_i \) = supply from node corresponding to customer location \( v_i \)

\( Q = \begin{cases} Q_A \text{ if vehicle is of type A} \\ Q_B \text{ if vehicle is of type B} \end{cases} \quad (Q_A: 2050, Q_B: 5200)

\( x_{ijk} = \begin{cases} 1, \text{ if vehicle } k \text{ travels directly from node } v_i \text{ to node } v_j \\ 0, \text{ otherwise} \end{cases} \quad (i, j) = 0,1,..,n \quad (k = 1,..,m)

\( y_{ik} = \begin{cases} 1, \text{ if node } v_i \text{ is serviced by vehicle } k \\ 0, \text{ otherwise} \end{cases} \quad (i, = 0,1,...,n) \quad (k = 1,..,m)

Variables \( x_{ijk} \) and \( x_{ijk} \) are only defined if \( q_i + q_j \leq Q \)

Objective Function

\[
\text{Min} \sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} x_{ijk} \quad (6)
\]

(Minimize total distance travelled (transportation costs))

\[
\text{Min} m \quad (7)
\]

(Minimize number of trips corresponding to the number of vehicles)

Constraints

\[
\sum_{i=0}^{n} q_i y_{ik} \leq Q_A \text{ or } Q_B \quad (k = 1,2, ..., m) \quad (8)
\]

(Prevent vehicles from carrying loads more than their capacity \( Q_A \) or \( Q_B \))

\[
\sum_{k=1}^{m} y_{ik} = \begin{cases} m, \quad (i = 0) \\ 1, \quad (i = 1,2,...,n) \end{cases} \quad (9)
\]

(Ensures that each vehicle leaves and arrives at the depot exactly once and each customer is served by one and only one vehicle)

\[
\sum_{i=0}^{n} x_{iok} = m \quad (10)
\]

(Ensures that the vehicles leave once at each node and \( m \) times at the depot \( o \))

\[
\sum_{j=0}^{n} x_{ojk} = m \quad (11)
\]

(Ensures that the vehicles arrive at each node once and leaves the depot \( o - m \) times)

\[
m = k \quad (12)
\]

(Ensures that the total number of trips is equal to the total number of vehicles)

\[
\sum_{i=0}^{n} \sum_{j=0}^{n} y_{ik} (t_{ij} + s_i) \leq T_k \quad (k = 1,2,..,m) \quad (13)
\]

(Ensures that the vehicles adhere to the time windows)

\[
e_i \geq E_i \quad (i, = 0,1,...,n) \quad (14)
\]

(Ensures that arrival time at any customer should not be before earliest time allowed for pickup)
\[ L_i \geq l_i \quad (i, = 0,1,...,n) \]  
(Ensures that arrival time at any customer should not be after latest time allowed for pickup)

\[ t_j \geq t_i + t_{ij} - (1 - x_{ijk}).T \quad (i, j = 0,1,...,n) \quad (k=1,2,...,m) \]  
(Ensures that arrival times between two customers are compatible)

\[ a_i \leq t_i < l_i \quad (i = 0,1,...,n) \]  
(Ensures the arrival time of a vehicle at a customer site to be within the customers earliest and latest arrival times)

\[ t_i \geq 0 \quad (i = 0,1,...,n) \]  
(Ensures that the arrival time of the vehicle at a customer location is always positive)

**Solution Approach based on Jaya Algorithm Based Clustering (Cluster First and Route Second)**

In this method, Jaya algorithm is first applied to form clusters and subsequently the optimum route is found by Savings Algorithm.

The principle idea behind Jaya is based on the concept that the solution obtained for a given problem should move towards the best solution and should avoid the worst solution.

X&Y coordinates of the centroid of each cluster is considered as subjects and each student corresponds to a set of pre-defined centroids. For every student, clusters are generated using the following logic: Each customer is allocated to the nearest centroid if it satisfies the time and capacity constraint. If not, it is allocated to the second nearest centroid, if not to the third nearest and the process repeats till all allocations are done. The fitness function is defined as total distance travelled by each candidate which is nothing but the total route distance.

**Figure 2: Cluster First – Route Second Logic**

*Steps in Application of Jaya + Savings Algorithm to Vehicle Routing Problem*
1. **Define Optimization Problem & Initialize**
   **Optimization Parameters:**
   - **Objective Function:** Minimize \( f(x) \): Minimize total distance travelled (proportional to transportation time and costs) and total number of vehicles.
   - **Population size** \( P_n \): 100 (Number of candidate solutions)
   - **Design variables** \( D_n \): 2
   - **Iterations:** 50
   - **No. of routes required to serve all customers** are calculated as follows (rounded to next integer):
   
   \[
   \text{Initial No. of Routes (Vehicles)} = \frac{\text{Total demand at all customers}}{\text{Vehicle capacity}}
   \]
   The number of routes is incremented by 1 in case we fail to obtain feasible solution.
   - **Lower and upper limits for the centroid values** are \( (LL, UL) \):
     - \( LL = [6.4 1.4 6.4 1.4 6.4 1.4 6.4 1.4] \)
     - \( UL = [18 26 18 26 18 26 18 26] \)

2. **Initialize the Population:**
   - Randomly create the population based on population size \( P_n \): 100 and 2 times the number of minimum routes.

3. **Identify Initial Solution:**
   - (Formation of Clusters & Route for First Iteration)
   - Determine best pair of centroids to form the clusters
   - **3.1 Formation of Clusters using the set of centroids**
     - **3.1.1 Generate clusters for the set of centroids given as input**
     - **3.1.2 Assign each customer to the nearest centroid (cluster) such that these satisfy the constraints viz., vehicle capacity, time window (travel time + loading time)**
     - **3.1.3 Generate network of clusters formed**
   - **3.2 Formation of Route for the given clusters**
     - **3.2.1.1 Initialize new route, ordered matrix and savings matrix**
     - **3.2.1.2 Calculate savings matrix and list the savings in ascending order in the ordered matrix**
     - **3.2.1.3 Calculate all other routes their distance, time and capacity for the same**
     - **3.2.1.4 Return the formed clusters and total minimum distance and maximum distance calculated for all routes**

4. **Generate new solution repeating step 3 till all the iterations are performed** \( P_n \): 100

4.1 **Modify the solution** based on best and worst solutions as
   \[
   X'j,k,i = (Xj,best,i - Xj,worst,i) - r2,j,i
   \]

4.2 **Select Best of the previous Two Solutions:**
   Accept the new solution if it is better than previous solution, else reject it and retain the previous solution.

The algorithm was run for 20, 50 and 100 population and 2 design variables. However, it was observed that the results were remaining the same after 12 iterations. On generation of each cluster, permutations of all possible routes were done to determine the optimum route within each cluster. The solution space available for Jaya Algorithm increases substantially. This gives rise to the possibility of better solutions. Figure 3 shows the number of iterations required to get the optimized solution.

![Figure 3: No. of Iterations vs Solution Quality](image)

**Proposed Solution**

- Determine monthly requirements for each component (from firm production plan)
- Calculate total volume to be collected from each supplier (for all components)
- Determine frequency of optimum pickup runs for local collection and number of trips required from Mumbai to manufacture.
- Solve the problem as VRP

**4. The optimal route formation (including the individual route distance and time) and the total distance and time travelled is shown in figure 4.**

![Figure 4: Formation of Routes & Final Solution](image)
RESULTS & FINDINGS

The benefits derived from the proposed method in terms of number of trips, transportation costs and savings is summarized in Table 5 below.

Table 5: Comparison of Existing & Proposed Methods

<table>
<thead>
<tr>
<th></th>
<th>Existing Method</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of Trips</td>
<td>98</td>
<td>120</td>
</tr>
<tr>
<td>Transportation Cost</td>
<td>Fixed Charge Per Route</td>
<td>(₹ 18 / km)</td>
</tr>
<tr>
<td>Calculations</td>
<td>₹ 1,60,825 / month</td>
<td>₹ 104,220 / month</td>
</tr>
<tr>
<td></td>
<td>₹ 19,29,900 / year</td>
<td>₹ 12,50,640 / year</td>
</tr>
<tr>
<td>Total Transportation Cost</td>
<td>₹ 19,36,725</td>
<td>₹ 32,74,860</td>
</tr>
<tr>
<td></td>
<td>Net Savings: ₹ 56,605 per month (₹ 6,79,260 per annum) [35.2 %]</td>
<td></td>
</tr>
</tbody>
</table>

CONCLUSION

The paper addresses a real-life case study for consumer durable manufacturer of optimization of routes for collection and transportation of components from suppliers to collection centre. A new hybrid metaheuristic based on Jaya Algorithm and Savings Algorithm is introduced to multi-objective capacitated vehicle routing problems with time constraints. This new method is based on cluster first and route second approach. This technique uses combination of Jaya heuristic (which explores a large number of cluster formations and hence enhances the possibility of better solutions) and Savings Algorithm with 2-Opt improvement heuristic. Computational results show the possibility of maintaining solution quality.

REFERENCES