Modeling a software platform for beehive placement optimization

Vladimir Milićević1*, Nemanja Zdravković1, Jovana Jović1, Dušan Jagličić2

1 Faculty of Information Technologies, Belgrade Metropolitan University, Tadeuša Košćuška 63, 11000 Belgrade
2 BeeWeb, Skerlićevo 7/3, 34000 Kragujevac

*Corresponding author: vladimir.milicevic@metropolitan.ac.rs

Received 3 December 2021; Accepted 4 April 2022

ABSTRACT
In this paper, we discuss the multi-depot vehicle routing problem with regard to mobile beekeeping. Considering that beekeepers usually do not have their own land plots rarely rely on external transporters and often rely on external land plots, we present a fully functional platform to pair beekeepers, transporters, and land plot owners. Two algorithms for calculating optimal mobile beekeeping routes are developed in order to increase honey yield and minimize transport costs. We show that the first algorithm, based on the mathematical model, creates optimal routes in order to transport and distribute beehives by a distributor, while minimizing transport costs. In addition, we consider an option for the case when beekeepers themselves want to handle the transport and use a single external land plot. We show that, for this case, the application of the second algorithm creates a "good enough" solution. This iterative algorithm routes beekeepers to the closest land plot that can handle their beehives. Furthermore, we present a fully functional web platform that applies these algorithms, enabling beekeepers, transporters, and land owners to register and use the platform in Serbia.

Keywords: beekeeping, supply chain, multi-depot vehicle routing problem, routing optimization, transport optimization.

1. Introduction
Beekeeping, as an agricultural venture, requires relatively low start-up costs, the ownership of minimum land space to stand a hive, very little labor, and other locally available inputs, and is an attractive economic pathway for further advancement (Carroll and Kinsella, 2013). Nevertheless, beekeeping and honey production are still considered inefficient, especially in underdeveloped countries, due to general problems beekeepers face. Beekeepers rely either on their own land plots, called apiaries, or have their own network of trusted partners to transport the beehives, mostly by themselves. Nowadays, bees also experience many different types of stresses such as the impact of parasitic mites, microsporidian pathogens and hundreds of pesticides applied in crops. One of the strongest stress factors affecting bees is long-distance transportation especially because of the little understanding of the effects of transportation on bees because no studies have ever been conducted to determine the physiological or behavioral changes induced by such stress (Kiheung et al., 2012). In addition, the honey supply chain is handled either by obsolete technologies or no use of technology at all. All these factors contribute to suboptimal honey production, high beehive relocation costs and non-uniform pollination.

The problem of the honey supply chain can be viewed as a multi-depot vehicle routing problem (MDVRP) (Montoya-Torres et al., 2015). The MDVRP is a generic name referring to a class of combinatorial optimization problems in which customers are to be...
served by a number of vehicles. The vehicles leave the depots, serve customers in the network and return to the depots after completion of their routes. Thereafter, a considerable number of variants have been considered: hard, soft and fuzzy service time windows, maximum route length, pickup and delivery, backhauls, etc. Solving the MDVRP is vital in the design of distribution systems in supply chain management. Ramos et al. (2020) considered a heterogeneous vehicle fleet with maximum routing time constraints, and proposed a two-commodity flow formulation. Mahmud and Haque (2019) provided a genetic algorithm approach to solving the MDVRP, which performed better when compared to other bio-inspired solutions.

Although the MDVRP problem for the supply chain has been extensively addressed in the literature, when it comes to beekeeping, there exist only a handful of papers. Pilati and Fontana (2018) examined the microeconomic model of the migratory beekeeper, indicating the benefits of migrating apiaries. They also pointed out the possible challenges, especially the policies regarding migratory beekeeping and biodiversity. In most of the reviewed papers, the problem has been modeled with deterministic parameters; few have modeled the real world uncertainties (Fritz and Schiefer, 2009). MDVRP needs to be complemented by the ability to make and implement sophisticated decisions in real-time in order to respond effectively to unforeseen events. The emergence of technologies and information systems allowing for seamless mobile and wireless connectivity between delivery vehicles and distribution facilities is paving the way for innovative approaches in addressing this requirement (Giaglis et al., 2004).

The goal of this paper is twofold: a) develop a mathematical model, and then a software solution for the MDVRP with regard to the mobile beekeeping problem in Serbia, and b) develop a platform which would utilize the solution for two distinct cases – when beekeepers want their beehives on multiple apiaries, and on a single apiary. We have, hence, developed a fully functional platform, called BeeWeb, which utilizes two algorithms for the MDVRP and geolocation in the backend, and provides a simple interface on the frontend, tackling all the problems beekeepers, transporters, and land plot owners have in mobile beekeeping.

The paper is organized as follows: Section 2 presents motivation for this work. Section 3 describes the mathematical model of the proposed solution. Section 4 presents the BeeWeb – Beekeeper – Apiary distance selection algorithm. Section 5 provides model evaluation results. Section 6 presents the software solution interface. Section 7 concludes the paper.

2. Motivation

The main goal of the BeeWeb Project is to apply a MDVRP solution in order to increase honey yield per beehive, decrease bee relocation costs, and increase apiary efficiency by uniform pollination. Apart from the Project’s general value in plant pollination, which ensures a better and healthier environment, there are three distinct groups of stakeholders: beekeepers, farmers, and honey-based product distributors. This Project aims to connect farmers growing honey-producing crops with beekeepers, and beekeepers with distributors. In addition, the Project aims to promote mobile beekeeping, which is not the dominant type of beekeeping in the Republic of Serbia. Stakeholders are to be connected by a special algorithm designed to reduce colony transport costs, taking into consideration available resources and distance.

By applying this method, it is possible to obtain higher honey yields while avoiding colony overpopulation, a common field problem. In addition, the Project would result in reduced transport costs, easier colony relocation procedures, and uniform crop pollination, and would motivate stationary beekeepers to switch to mobile beekeeping to increase their produce.

In order to meet the Project’s requirements, the following goals have been identified:

- Define the set PC, whose elements are beekeepers, each having a number of beehives,
- Define the set PAR, whose elements are land plots of specific areas on which beehives can be deployed.
- Depending on the type of crop (sunflower, meadow flowers, acacia…) define a coefficient which equals the number of beehives per 1 ha.
- Determine an arranging algorithm for the elements of PC onto the elements of PAR, minimizing the travel cost.

When researching suitable algorithms, operational research methods first came to mind. If beehives are treated as a load to be transported to the locations defined by the set PAR, by applying a transport problem, either standardly or using linear programming, one can obtain the needed algorithm to map elements of PC and PAR. These techniques have been applied in numerous studies (Carroll and Kinsella, 2013; Kiheung et al., 2012; Montoya-Torres et al., 2015; Ramos et al., 2020); however, to the authors’ knowledge, the problem of increasing apiary efficiency by minimizing beehive transport to a set of offered locations has not been addressed in the available literature.

To meet the identified goals, the authors of this paper have developed an algorithm to enable the pairing of the elements of PC and PAR with linear programming for this specific case of the transport problem.

Given the requirements of end users, in this case beekeepers, certain limiting factors have been raised. Beekeepers that mostly rely on their own transport are not eager to distribute their beehives on a large number of farms. This, in turn, has led to the development of a software solution based on an alternative algorithm to pair beekeepers and farmers, taking into account these additional limiting factors.

Additionally, the system can be integrated into a GIS system for mapping beekeepers and different types of crops (vegetation) (Lemenkova, 2021).

3. Model development

During the development of the software model, two algorithms have been created, implemented, tested and compared, and relevant conclusions have been made.
3.1. Algorithm 1 – Mathematical model

We start by defining two sets \( PC = \{ X_1, X_2, \ldots, X_n \} \), \( PAR = \{ P_1, P_2, \ldots, P_m \} \), which represent the number of hives owned by beekeepers, and areas of available land on plots, respectively. An integer variable defines the number of beekeeper’s beehives deployed on the land plot. The number of beehives that can be deployed on 1 ha of land, depending on the type of apiary flower stand, is defined. The distance from a beekeeper to the land plot is defined by a variable, calculated from Google Maps or using well-known mathematical calculations.

Our goal is to define a function to minimize transport costs:

\[
\begin{align*}
\min \{ x_1d_1 + x_2d_2 + \ldots + x_TD_T + x_2d_21 + x_2d_22 + \ldots + x_Td_Tm + x_2d_2T \\
+ x_2d_21 + \ldots + x_Td_Tm \}.
\end{align*}
\]

(1)

The next step is to introduce an equation system to describe the limiting factors during the analysis.

\[
\begin{align*}
x_1 + x_2 + \ldots + x_4 \leq \alpha P_1 \\
x_2 + x_3 + \ldots + x_6 \leq \alpha P_2 \\
\vdots \\
x_m + x_2m + \ldots + x_m \leq \alpha P_n
\end{align*}
\]

(2)

\[
\begin{align*}
x_1 + x_2 + \ldots + x_4 = X_1 \\
x_2 + x_2 + \ldots + x_6 = X_2 \\
\vdots \\
x_m + x_2m + \ldots + x_m = X_n
\end{align*}
\]

(3)

It is necessary to introduce an additional limiting factor to obtain correct results:

\[
\sum_{i=1}^{n} X_i = \alpha \sum_{j=1}^{m} P_j
\]

(4)

If condition (4) is not fulfilled, e.g. the total number of beehives that can be deployed in PAR is greater than the total number of beehives that beekeepers from PC hold, it is necessary to inset a fictive beekeeper to satisfy condition (4).

A mathematical model to deploy beehives, while minimizing transport costs subject to the limiting factors, is therefore introduced. However, by directly contacting the beekeepers, a new constraint has emerged – most of the beekeepers are not eager to distribute their own beehives on multiple land plots. Therefore, this problem has to be addressed by:

- Acknowledging the additional constraints presented by the beekeepers;
- Respecting the proposed requirements with minimal tolerance;
- Replacing the optimal solution with the “good enough” solution.

This, in turn, means that the transportation problem should be approached heuristically, as presented in numerous papers (Mahmud and Haque, 2019; Pilati, and Fontana, 2018; Fritz and Schiefer, 2009; Giaglis et al., 2004; Otonkue et al., 2009; Khan, 2014; Ghazali et al., 2019).

3.2. Algorithm 2 – BeeWeb alternative algorithm

The following pseudocode describes the alternative algorithm.

I. Create the class Pcelari with the following:

- int lok_x;
- int lok_y;
- int id;
- boolean slobodan;

II. Create the class Parcele with the following:

- int kosnica_po_parceli; // kultura (broj_kosnica / ha) * povrsina_kosnice
- int lok_par_x;
- int lok_par_y;
- int id_par;

III. Create the main class Uparivanje:

a) Create an array of objects from the class Pcelari: pc[i], i = 1, …, n;

b) Sort the array pc[i] in a descending order by beehive number;

c) Create an array of objects from the class Parcele: par[i], i = 1, …, m;

d) Sort the array par[i] in an ascending order by the number of beehives it can take;

e) Define the boolean variable uparen with initial value FALSE;

f) Set the value of the variable slobodan of every element of the Pcelari array to TRUE;

IV.

a) Run a loop within the Uparivanje class with the counter i = 1, …, n, where i is the counter of pc[i];

b) Uparen = FALSE;

c) In every following iteration find the distance between the current beekeeper and all land plots defined by the par[i] array using Google maps or mathematical computations;

d) Create a new array dist[j], j = 1, …, m, sorted in an ascending order.

e) Navigating in the nested loop by counter j, where j is the counter of the distance array dist[j] between a beekeeper and a land plot, find the location with the shortest distance meeting the following requirement:

pc[i].broj_kosnica<= par[j].kosnica_po_parceli;

f) If the land plot is found and the status of the current beekeeper object is TRUE, when

- uparen = TRUE;
- slobodan = FALSE;
- save the pairing result (id, id_par)
- Reduce the number of beehives available by the number of unavailable beehives until the value of kosnica_po_parceli is not less or equal than the number of beehives of the last beekeeper in the array (in that case remove the land plot from any further searches).
4. **Beeweb – Beekeeper – Apiary Distance Selection Algorithm**

To determine the set of distances from a beekeeper to an apiary, we used two approaches:

1. Calculating the straight-line distance with the well-known mathematical formula for distance between two points on a plane

   \[ d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \]  

   Where the beekeeper and apiary positions are respectively determined by the set of coordinates \((x_1, y_1)\) and \((x_2, y_2)\).

2. Applying tools and algorithms that are available online and which are relatively easy to use:

   - a) Google map distance locator (Kumar and Panneerselvam, 2012), where a user can determine the start and destination points using a mouse, and the application returns a straight-line distance, as shown in Fig. 1 on the next page.
   - b) Distance calculator (You, Hsieh, Chen and Lee, 2013) which also relies on Google Maps, where a user can enter the start and destination points in the dedicated text boxes, resulting in a straight-line distance between those two points, as shown in Fig. 2.
   - c) ScribbleMaps (Tarantilis, Spinellis, and Gendreau, 2005) distance calculator, where a user can click on two points on a map as well. The ScribbleMaps tool is shown in Fig. 3.
   - d) The SAS 9.2 tool (Zdeb, 2010) which uses Vincenty’s formulae in its functions ZIPLYDISTANCE and GEODIST to determine the geodesic distance. Before applying the SAS 9.2 tool, these distances were determined using the haversine formula. Both formulas rely on straight/line distances and in most cases the accuracy would suffice. However, distances that cannot always be represented by straight lines are distances travelled by cars. These distances can easily be obtained using Google Maps and therefore do not present any limiting factor. If there exists a large number of location pairs, using the FILENAME and the URL method within the SAS 9.2 tool can be used multiple times to access Google Maps, in order to obtain the distance and travel time each time we access a location.
Figure 2. Map Developers distance calculator tool

Figure 3. Scribble Maps distance calculator tool

The choice of distance calculator tools ultimately fell on the mathematical model described in IV.1, which gave almost identical results when compared to the other methods mentioned. In addition, one of the reasons was the ease of implementing this model. On the other hand, Google has limited the free use of its Maps services to 25000 per month per user. This, in turn, means that a website which has more than 800 map views per day cannot use Google Maps for free. The cost of Google Maps Premium licenses, which start at USD 10000 per year, was a significant limiting factor as well (https://developers.google.com/maps/pricing-and-plans/, 2021).

5. Results and discussions

Both models described in Sections III and IV were successfully tested. The following subsections present the results.

5.1. Algorithm 1 – Mathematical model – applying linear programming

In the first step, it is necessary to include the lists of available beekeepers and farmers (land plots). Both lists are shown in Tables 1 and 2, respectively.
Table 1.
List of beekeepers

<table>
<thead>
<tr>
<th>ord.</th>
<th>Country</th>
<th>Region</th>
<th>City</th>
<th>Number of Beehives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Serbia</td>
<td>Beograd</td>
<td>Barajevo</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>Serbia</td>
<td>Kragujevac</td>
<td>Baljkovac</td>
<td>345</td>
</tr>
<tr>
<td>3</td>
<td>Serbia</td>
<td>Arilje</td>
<td>Radoševo</td>
<td>54</td>
</tr>
<tr>
<td>4</td>
<td>Serbia</td>
<td>Bač</td>
<td>Vajska</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>Serbia</td>
<td>Mali Zvornik</td>
<td>Culine</td>
<td>213</td>
</tr>
<tr>
<td>6</td>
<td>Serbia</td>
<td>Ljig</td>
<td>Lalinci</td>
<td>23</td>
</tr>
<tr>
<td>7</td>
<td>Serbia</td>
<td>Leskovac</td>
<td>Bričevje</td>
<td>76</td>
</tr>
<tr>
<td>8</td>
<td>Serbia</td>
<td>Vladimirici</td>
<td>Mehovine</td>
<td>128</td>
</tr>
<tr>
<td>9</td>
<td>Serbia</td>
<td>Žabalji</td>
<td>Gospodinci</td>
<td>212</td>
</tr>
<tr>
<td>10</td>
<td>Serbia</td>
<td>Vršac</td>
<td>Uljma</td>
<td>76</td>
</tr>
<tr>
<td>11</td>
<td>Serbia</td>
<td>Vrnjačka Banja</td>
<td>Vukušica</td>
<td>45</td>
</tr>
<tr>
<td>12</td>
<td>Serbia</td>
<td>Zaječar</td>
<td>Halovo</td>
<td>56</td>
</tr>
<tr>
<td>13</td>
<td>Serbia</td>
<td>Lapovo</td>
<td>Lapovo</td>
<td>188</td>
</tr>
<tr>
<td>14</td>
<td>Serbia</td>
<td>Čičevac</td>
<td>Pojate</td>
<td>218</td>
</tr>
</tbody>
</table>

Table 2.
List of farmers (land plots)

<table>
<thead>
<tr>
<th>ord.</th>
<th>Country</th>
<th>Region</th>
<th>City</th>
<th>Crop</th>
<th>Hectare</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Serbia</td>
<td>Zrenjanin</td>
<td>Banatsko Višnjičevo</td>
<td>sunflower</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>Serbia</td>
<td>Zrenjanin</td>
<td>Elemir</td>
<td>sunflower</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>Serbia</td>
<td>Zrenjanin</td>
<td>Hetin</td>
<td>sunflower</td>
<td>450</td>
</tr>
<tr>
<td>4</td>
<td>Serbia</td>
<td>Šećanj</td>
<td>Jaša Tomić</td>
<td>sunflower</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>Serbia</td>
<td>Šećanj</td>
<td>Konak</td>
<td>sunflower</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>Serbia</td>
<td>Šećanj</td>
<td>Krajšinik</td>
<td>sunflower</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Serbia</td>
<td>Šećanj</td>
<td>Sutjeska</td>
<td>sunflower</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>Serbia</td>
<td>Nova Črma</td>
<td>Vojvoda Stepa</td>
<td>sunflower</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>Serbia</td>
<td>Nova Črma</td>
<td>Radajčevo</td>
<td>sunflower</td>
<td>80</td>
</tr>
<tr>
<td>10</td>
<td>Serbia</td>
<td>Nova Črma</td>
<td>Aleksandrovo</td>
<td>sunflower</td>
<td>230</td>
</tr>
<tr>
<td>11</td>
<td>Serbia</td>
<td>Nova Črma</td>
<td>Toba</td>
<td>sunflower</td>
<td>120</td>
</tr>
<tr>
<td>12</td>
<td>Serbia</td>
<td>Kiklenda</td>
<td>Banatsko Veliko Selo</td>
<td>sunflower</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td>Serbia</td>
<td>Kiklenda</td>
<td>Rusko Selo</td>
<td>sunflower</td>
<td>90</td>
</tr>
<tr>
<td>14</td>
<td>Serbia</td>
<td>Kiklenda</td>
<td>Idoš</td>
<td>sunflower</td>
<td>66</td>
</tr>
</tbody>
</table>

In the next step, the distance between the beekeepers and the farmers is calculated using the methods from Section IV, and a distance matrix is formed, as shown in Fig. 4.

The parameters $d_{ij}$ represent the distance values inserted in the matrix. The goal of the model is to minimize the function $\sum \sum (x_{ij}d_{ij})$, where $x_{ij}$ is the number of beehives owned by beekeeper $i$ transported to the location owned by farmer $j$. With the goal and constraints in mind, the model equations are formed, and are passed to the Microsoft Access LP solver tool. When testing, all land plots with sunflowers were used, and for that crop, a deployment constraint of 4 beehives per 1ha is applied.

By clicking on the Solve button, the model is processed and the results are shown in Figure 5.
5.2. Algorithm 2 – Alternative software model

A software solution is implemented according to the constraint one beekeeper – one apiary. Two .xlsx documents are used as an input to the software solution, which correspond to the lists of beekeepers and farmers shown in Tables 1 and 2, respectively. Figure 6 shows the output of the software solution which pairs the beekeepers with the apiaries.

5.3. Algorithm 2 – Additional Considerations

The optimal result gives a total distance sum \( R_1 = 2260.1\) km, while the alternative software model gives a total distance of \( R_2 = 2359.27\) km. The difference between the model results, as well as the individual farmer distances are respectfully calculated as

\[
R = R_2 - R_1 = 99.16\ km \tag{6}
\]

and

\[
\Delta R = \frac{99.16}{14} = 7.08\ km \tag{7}
\]

In the test case, the optimal solution gives a total distance shorter by 99.16 km per direction. By applying this model, on average, the beekeepers have traveled 7.08 km less than by the software model calculation. This is deemed a small deviation, and therefore the implemented software model can indeed be considered a good enough applicable solution. It is worth noting the software solution is very efficient in processing the implemented model.

6. Software Solution Interface

A software model for BeeWeb has been developed as a web platform available at http://www.beeweb.co/srb. Upon opening the website on a browser, the user is greeted with a page as shown in Figure 7.
By scrolling, a user can see links dedicated to beekeepers, farmers, and transporters, as shown in Figure 8.

The beekeepers, farmers, and transporters can register either individually or as a group. Before the initial login, the user must register, while all existing users can login using the login form. Different user categories can access the pricing information for using BeeWeb, as shown in Fig. 9 on the next page. Both free and paid options are available. The pairing between the existing beekeepers and farmers is shown in Fig. 6.
7. Conclusions

We have identified two possible directions to solving beekeeper routing with the goal of increasing the efficiency of apiaries and minimizing transport costs. For the case when beehives are to be transported by a transporter, i.e., the beekeeper is not interested in the deployment of beehives, the optimal solution is given with Algorithm 1. Using this method, minimal total costs of transport to the apiaries are obtained.

Conversely, when beehives are to be transported by beekeepers themselves, who are not interested in deploying their beehives, we can only discuss a good enough solution by applying Algorithm 2. For that case, beekeepers with the greatest number of beehives are routed to the nearest location that can handle the number of beehives. This process is repeated iteratively until the beekeeper with the least number of beehives is paired.

Acknowledgment

The software solution is the result of work on Project FP-7, Fractals, BeeWeb - FP7 project No. 632874.

Declaration of competing interest

Authors do not have personal or financial relationships with other people or organizations that could improperly influence (bias, non-compliance with the academic code) their work.

References


