UDC 638.1:004.4
Acta Agriculturae Serbica, 27 (53), 39–48, 2022
doi: 10.5937/AASer2253039M
Publisher: Faculty of Agronomy Čačak
www.afc.kg.ac.rs

Modeling a software platform for beehive placement optimization

Vladimir Milićević[1*], Nemanja Zdravković[1], Jovana Jović[1], Dušan Jagličić[2]

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

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

Received 3 December 2021; Accepted 4 April 2022

A B S T R A C T

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.

И З В О Д

У овом раду биће говора о специјалном случају проблема сложеног рутирања са фокусом на мобилно пчеларство. С обзиром на то да пчелари најчешће немају своје парцеле и ретко се ослањају на спољне транспортере али и често на спољне парцеле, представљамо потпуно функционалну платформу за упаривање пчелара, транспортера и власника парцела. Развијена су два алгоритма за прорачун оптималних пчеларских рута у циљу повећања приноса меда и минимизирања транспортних трошкова. Показујемо да први алгоритам, базиран на математичком моделу, креира оптималне руте за транспорт и дистрибуцију кошница од стране дистрибутера, минимизирајући транспортне трошкове. Поред тога, разматрамо опцију за случај када пчелари сами желе да се баве транспортом и да користе једну спољну парцелу. Показујемо да се за овај случај применом другог алгоритма ствара „довољно добро" решење. Овај итеративни алгоритам усмерава пчеларе до најближе парцеле која може да рукује њиховим кошницама.

Поред тога, представљамо потпуно функционалну веб платформу која примењује ове алгоритме, омогућавајући пчеларима, превозницима и власницима земљишта да се региструју и користе платформу у Србији.

Кључне речи: пчеларство, ланац снабдевања, проблем сложеног рутирања, оптимизација рутирања, оптимизација транспорта.

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 $\boldsymbol{PC} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_n}\}$, $\boldsymbol{PAR} = \{\boldsymbol{P_1}, \boldsymbol{P_2}, \dots, \boldsymbol{P_m}\}$, 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 1ha 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:

$$\min\{x_{11}d_{11} + x_{12}d_{12} + \ldots + x_{1m}d_{1m} + x_{21}d_{21} + x_{22}d_{22} + \ldots + x_{2m}d_{2m} + \ldots + x_{n1}d_{n1} + x_{n2}d_{n2} + \ldots + x_{nm}d_{nm}\}. \quad (1)$$

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

$$\begin{aligned} x_{11} + x_{21} + \ldots + x_{n1} &\le \alpha P_1 \\ x_{12} + x_{22} + \ldots + x_{n2} &\le \alpha P_2 \\ &\vdots \\ x_{1m} + x_{2m} + \ldots + x_{nm} &\le \alpha P_m \end{aligned} \quad (2)$$

$$\begin{aligned} x_{11} + x_{12} + \ldots + x_{1n} &= X_1 \\ x_{21} + x_{22} + \ldots + x_{2n} &= X_2 \\ &\vdots \\ x_{n1} + x_{n2} + \ldots + x_{nm} &= X_n \end{aligned} \quad (3)$$

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

$$\sum_{i=1}^{n} X_i = \alpha \sum_{j=1}^{m} P_j \quad (4)$$

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);

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

g) If no suitable land plot has been found in the current iteration:
- Divide the number of the current beekeeper's hives into two new beekeepers;
- The newly formed beekeepers are given the first two indexes while the rest are shifted by one spot;
- Return to step e) and
- Repeat the process for the (n+1)th member of the array pc[i];

h) Continue pairing while there are unpaired elements of pc[i];

V. List all the pairing results: (pc[i], par[j]), i = 1, … , n, j = 1, … , m.

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

$$d = \sqrt{\left(x_1^2 - x_2^2\right) + \left(y_1^2 + y_2^2\right)} \qquad (5)$$

where the beekeeper and apiary positions are respectively determined by the set of coordinates $(\boldsymbol{x_1}, \boldsymbol{y_1})$ and $(\boldsymbol{x_2}, \boldsymbol{y_2})$.

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 ZIPCITYDISTANCE 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 1. Google Maps distance calculator

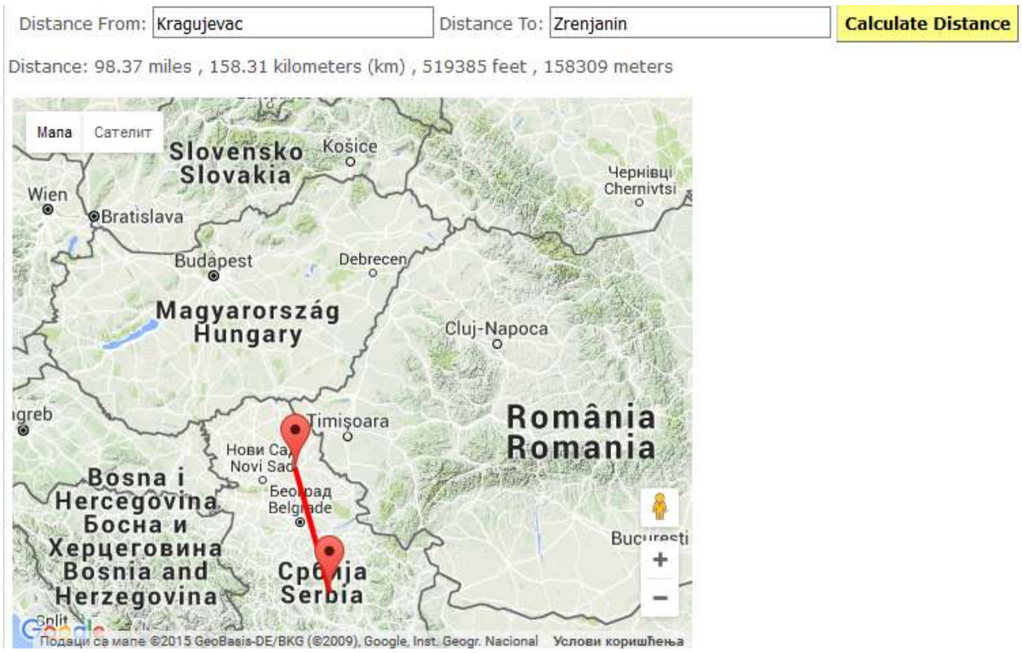

Figure 2. Map Developers distance calculator tool

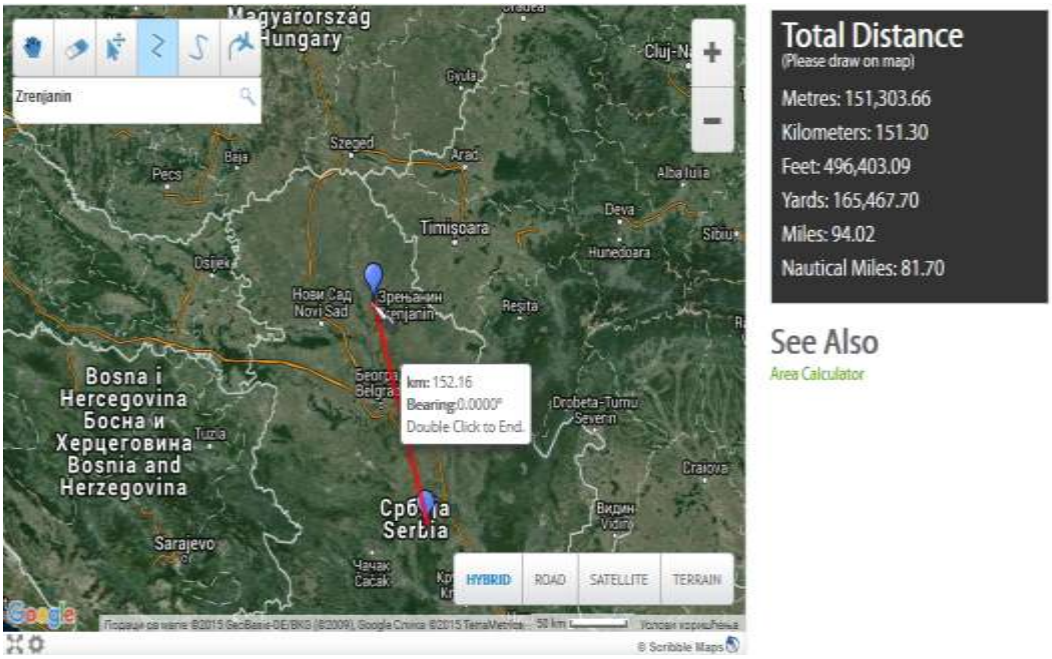

Figure 3. Scribble Maps distance calculator tool

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

ord.	Country	Region	City	Number of Beehives
1	Serbia	Beograd	Barajevo	34
2	Serbia	Kragujevac	Baljkovac	345
3	Serbia	Arilje	Radoševo	54
4	Serbia	Bač	Vajska	72
5	Serbia	Mali Zvornik	Culine	213
6	Serbia	Ljig	Lalinci	23
7	Serbia	Leskovac	Bričevlje	76
8	Serbia	Vladimirci	Mehovine	128
9	Serbia	Žabalj	Gospođinci	212
10	Serbia	Vršac	Uljma	76
11	Serbia	Vrnjačka Banja	Vukušica	45
12	Serbia	Zaječar	Halovo	56
13	Serbia	Lapovo	Lapovo	188
14	Serbia	Ćićevac	Pojate	218

Table2.
List of farmers (land plots)

ord.	Country	Region	City	Crop	Hectare
1	Serbia	Zrenjanin	Banatsko Višnjićevo	sunflower	23
2	Serbia	Zrenjanin	Elemir	sunflower	45
3	Serbia	Zrenjanin	Hetin	sunflower	450
4	Serbia	Sečanj	Jaša Tomić	sunflower	34
5	Serbia	Sečanj	Konak	sunflower	23
6	Serbia	Sečanj	Krajišnik	sunflower	20
7	Serbia	Sečanj	Sutjeska	sunflower	50
8	Serbia	Nova Crnja	Vojvoda Stepa	sunflower	45
9	Serbia	Nova Crnja	Radojevo	sunflower	80
10	Serbia	Nova Crnja	Aleksandrovo	sunflower	230
11	Serbia	Nova Crnja	Toba	sunflower	120
12	Serbia	Kikinda	Banatsko Veliko Selo	sunflower	15
13	Serbia	Kikinda	Rusko Selo	sunflower	90
14	Serbia	Kikinda	Iđoš	sunflower	66

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.

It is necessary to include the lists of available beekeepers and farmers (land plots). Both lists are shown in Tables 1 and 2, respectively.

Distance / Beekeepers	Farmers: Banatsko Višnjićevo	Elemir	Hetin	Jaša Tomić	Konak	Krajišnik	Sutjeska	Vojvoda Stepa	Radojevo	Aleksandrovo	Toba	Banatsko Veliko Selo	Rusko Selo	Iđoš
Barajevo	95.96	95.25	120.24	94.20	89.96	98.44	90.00	120.07	126.13	114.14	121.47	136.61	125.85	135.51
Baljkovac	166.75	173.33	198.68	161.65	157.98	168.06	160.15	190.51	195.33	185.86	194.35	209.15	198.68	211.88
Radoševo	207.26	200.62	231.59	206.26	201.98	210.02	201.59	230.97	237.39	224.61	231.12	246.09	235.40	241.48
Vajska	125.22	94.28	134.14	134.89	133.01	128.99	127.01	129.64	135.65	123.77	120.52	125.92	121.64	106.34
Culine	174.05	154.63	196.02	177.83	173.73	177.87	170.57	193.59	200.82	189.71	189.71	202.77	193.31	191.15
Lalinci	147.01	143.89	171.29	145.25	141.01	149.51	141.07	171.03	177.18	164.99	172.07	187.19	176.42	184.71
Bričevlje	315.32	328.39	334.76	307.85	305.02	315.53	308.71	337.06	340.27	333.97	343.12	356.77	347.18	363.78
Mehovine	115.49	99.23	138.26	118.64	114.48	119.19	111.62	136.23	143.36	129.11	133.43	147.14	137.31	138.42
Gospođinci	55.07	24.67	68.33	64.35	62.29	59.01	56.34	64.47	71.42	57.60	57.47	68.17	60.16	54.69
Uljma	107.89	80.57	76.12	49.80	47.59	57.91	52.36	78.59	81.61	76.15	85.43	98.39	89.26	107.89
Vukušica	200.71	204.51	223.91	196.29	192.47	202.30	194.19	224.73	229.87	219.71	227.91	242.89	232.28	244.34
Halovo	212.92	231.87	227.62	203.60	201.88	211.92	206.90	230.90	232.39	232.39	229.64	250.49	242.39	262.11
Lapovo	145.28	155.41	166.94	139.22	135.82	146.17	138.62	168.47	172.78	164.36	173.18	187.63	177.43	192.23
Pojate	199.07	210.86	219.63	192.21	189.11	199.59	192.41	221.56	225.32	217.97	226.99	241.06	231.15	246.93
Fiktivni	257.43	257.85	281.18	253.88	249.90	259.37	251.08	281.62	287.15	276.17	283.94	299.05	288.32	298.38

Figure 4. Distance matrix

The parameters d_{ij} represent the distance values inserted in the matrix. The goal of the model is to minimize the function $\sum_i \sum_j (x_{ij}, d_{ij})$, 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.

Matching / Beekeepers	Banatsko Višnjićevo	Elemir	Hetin	Jaša Tomić	Konak	Krajišnik	Sutjeska	Vojvoda Stepa	Radojevo	Aleksandrovo	Toba	Banatsko Veliko Selo	Rusko Selo	Idoš	[illegible]
Barajevo	0	0	0	0	0	0	34	0	0	0	0	0	0	0	34
Baljkovac	0	0	345	0	0	0	0	0	0	0	0	0	0	0	345
Radoševo	54	0	0	0	0	0	0	0	0	0	0	0	0	0	54
Vajska	0	72	0	0	0	0	0	0	0	0	0	0	0	0	72
Culine	0	0	0	0	0	0	0	0	0	213	0	0	0	0	213
Lalinci	0	0	0	0	23	0	0	0	0	0	0	0	0	0	23
Bričevlje	0	0	0	0	0	0	0	0	0	76	0	0	0	0	76
Mehovine	0	0	0	0	0	0	128	0	0	0	0	0	0	0	128
Gospođinci	0	0	0	0	0	0	0	0	0	0	0	0	0	212	212
Uljma	0	0	0	0	0	76	0	0	0	0	0	0	0	0	76
Vukušica	0	0	0	0	45	0	0	0	0	0	0	0	0	0	45
Halovo	0	0	0	56	0	0	0	0	0	0	0	0	0	0	56
Lapovo	0	0	188	0	0	0	0	0	0	0	0	0	0	0	188
Pojate	0	0	0	0	0	0	0	0	0	218	0	0	0	0	218
Fiktivni	38	108	1267	80	24	4	38	180	320	413	480	60	360	52	3424
[illegible]	23	45	450	34	23	20	50	45	80	230	120	15	90	66	
[illegible]	92	180	1800	136	92	80	200	180	320	920	480	60	360	264	[illegible]

SUM DISTANCE R1:	2260,11

Figure 5. Beekeeper – farmer pairing results

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.

Povezivanje

Poljoprivrednik	Hektara	Mesto	Pčelar	Broj košnica	Mesto	Udaljenost
3	450	Hetin	2	345	Baljkovac	190.68
3	450	Hetin	14	218	Pojate	219.63
3	450	Hetin	5	213	Culine	196.02
3	450	Hetin	9	212	Gospodinci	68.33
7	50	Sutjeska	13	188	Lapovo	138.82
3	450	Hetin	8	128	Mehovine	138.26
4	34	Jaša Tomić	10	76	Uljma	49.30
5	23	Konak	7	76	Bričevlje	306.02
2	45	Elemir	4	72	Vajska	94.28
4	34	Jaša Tomić	12	56	Halovo	203.60
3	450	Hetin	3	54	Radoševo	231.59
3	450	Hetin	11	45	Vukušica	223.91
3	450	Hetin	1	34	Barajevo	120.24
3	450	Hetin	6	23	Lalinci	171.29

Sum Distance: 2359.27 km

Figure 6. Beekeeper–farmer pairing results

The result in Fig. 6 is considered good enough and gives a total distance R2 (the distance traveled by all beekeepers in one direction) of 2359.27 km.

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

$$\boldsymbol{R = R_2 - R_1 = 99.16\ km} \quad (6)$$

and

$$\frac{\Delta R}{n} = \frac{99.16}{14} = 7.08\,\text{km} \quad (7)$$

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.

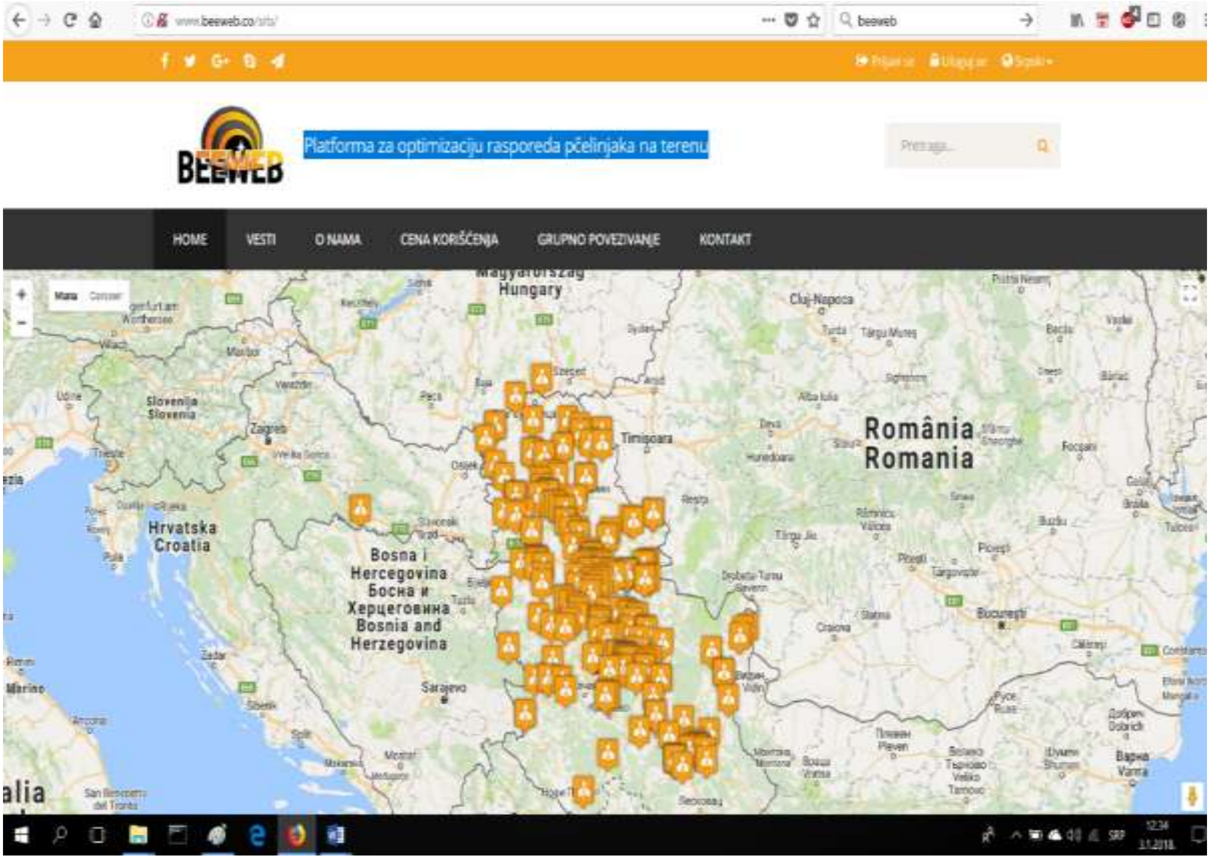

Figure 7. BeeWeb homepage

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.

Figure 8. Links dedicated to beekeepers, farmers, and transporters

Figure 9. BeeWeb pricing information

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 at multiple apiaries, 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

Bekkerman, A. (2013). Going the Distance: Google Maps Capabilities in a Friendly SAS Environment. *Proceedings of the 2013 WSUS conference*, 2013, 1–9.

Carroll, T., Kinsella, J. (2013). Livelihood improvement and smallholder beekeeping in Kenya: the unrealised potential. *Development in Practice*, 23(3), 332–345.

Fritz, M., Schiefer, G. (2009). System dynamics and innovation in food networks. Proceedings of the 2nd International European Forum on System Dynamics and Innovation in Food Networks, 18–22

Ghazali, Z., Majid, M.A., Shazwani, M. (2012). Optimal Solution of Transportation Problem Using Linear Programming: A Case of a Malaysian Trading Company. *Journal of Applied Sciences*, 12, 2430–2435.

Giaglis, G., Minis, I., Tatarakis, A., Zeimpekis. V. (2004). Minimizing logistics risk through real-time vehicle routing and mobile technologies. *International Journal of Physical Distribution & Logistics Management*, 34(9), 749–764.

Khan, M.A. (2014). Transportation Cost Optimization Using Linear Programming, International Conference on Mechanical, Industrial and Energy Engineering 2014, 25–26 December, 2014, Khulna, Banglades, 1–5

Kiheung, A., Xie, X., Riddle, J., Pettis, J., Huang Z.Y. (2012). Effects of long-distance transportation on honey bee physiology. *Psyche: A Journal of Entomology,* Volume 2012, 193029.

Kumar, S.N., Panneerselvam, S,R. (2012). A Survey on the Vehicle Routing Problem and Its Variants. *Intelligent Information Management*, 4, 66–74.

Laporte, G. (1992). The traveling salesman problem: Overview of algorithms. *European Journal of Operational Research*, 59, 231–247.

Laporte, G. (1992). The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59, 345–358.

Lemenkova, P. (2021). Evaluating land cover types from Landsat TM using SAGA GIS for vegetation mapping based on ISODATA and K-means clustering. *Acta Agriculturae Serbica*, 26(52), 159–165.

Mahmud, N., Haque, M.M. (2019). Solving Multiple Depot Vehicle Routing Problem (MDVRP) using Genetic Algorithm. *Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1–6.

Montoya-Torres, J.R., Franco, J.L. Isaza, S.N., Jiménez, H.F., Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79, 115–129.

Otonkue, A.D., Edu, B.E., Esang, B.E,A.E. (2011). Linear Programming: A Practical Approach to Transportation Cost Problems. *IMS Manthan*, 6(2), 10–14.

Pilati, L., Fontana, P. (2020). Sequencing the Movements of Honey Bee Colonies between the Forage Sites with the Microeconomic Model of the Migratory Beekeeper, Beekeeping – New Challeges – Intech Open Book Series, ISBN 978-1-83880-476-3,

Ramos, T.R.P., Gomes, M.I., Póvoa, A.P.B. (2020). Multi-depot vehicle routing problem: a comparative study of alternative formulations. *International Journal of Logistics Research and Applications*, 23(2), 103–120.

Reeb, J., Leavengood, S. (2002). Transportation Problem: A Special Case for Linear Programming Problems. Operation Research EM 8779 - June 2002, Oregon State University, 2–36.

Tarantilis, C.D., Spinellis, D., Gendreau, M. (2005). Advanced Heuristics in Transportation and Logistics. *IEEE Intelligent Systems*, 20(4), 16–18.

You, P., Hsieh, Y., Chen, T., Lee, Y. (2013). A Heuristic Approach for Assembly Scheduling and Transportation Problems with Parallel Machines. *iBusiness*, 5(1), 27–30.

Zdeb, M. (2010). Driving Distances and Times Using SAS® and Google Maps. *SAS Global Forum 2010, Paper 050-2010, 1–8*.

http://www.daftlogic.com/projects-google-maps-distance-calculator.htm (accessed: 12 October 2021)

http://www.mapdevelopers.com/distance_from_to.php (accessed: 12 October 2021)

http://www.scribblemaps.com/tools/distance-calculator (accessed: 12 October 2021)

https://developers.google.com/maps/pricing-and-plans/ (accessed: 18 October 2021)

https://developers.google.com/maps/usagelimits/ (accessed: 18 October 2021)