1. Introduction

A Portuguese Healthcare Local Network (HLN) is responsible for providing care to the population living in a large municipality of the Porto District, with around 175 thousand inhabitants. Simultaneously, the HLN is also the direct reference for users from other cities. Thus, there are 318 thousand users to whom HLN provides health care.

HLN's administrative headquarters is in a hospital (HPT), where its transversal services are also located. Primary Healthcare (PHC) at HLN is organised in the Healthcare Centers Clusters (ACES), which groups the functional units. The HPT has its object to transport services for goods between its ten functional units (facilities) among the many transversal services provided. Also, the functional units have specific demands for goods and patients. Tables 1, 2, and 3 present, respectively, the facilities and HPT (DC) coordinates, the particular demand for goods (m3), and the specific demand for patients (registered users).

Table 1 - Facilities and DC location positions

	Coordinate 1	Coordinate 2
DC	41,18186	-8,66340
F1	41,21560	-8,67583
F2	41,22437	-8,69788
F3	41,25228	-8,70513
F4	41,19442	-8,69733
F5	41,18651	-8,67994
F6	41,19157	-8,65760
F7	41,19836	-8,61148
F8	41,20246	-8,63212
F9	41,20301	-8,63988
F10	41,18928	-8,68655

Source: From Google Maps

Table 2 – Demand for goods (m3)

	Volume
DC	0
F1	1.0
F2	1.0
F3	0.5
F4	0.7
F5	1.0
F6	0.5
F7	0.3
F8	2.0
F9	2.0
F10	1.9

Table 3 - Number of registered frequent users

	Users
DC	16783
F1	3026
F2	11506
F3	10448
F4	10453
F5	14262
F6	17608
F7	12493
F8	16449
F9	9682
F10	0

Besides that, the maximum value available to supply the facilities is 58 152,00 €/ year, which includes vehicle, maintenance, two drivers, TMS, and other operational costs. This value also consists of a distance of up to 65 Km/day with an approximate cost of 2,50 €/Km. Also, due to the type of vehicle, there is a maximum available volume of 12 m3. Finally, Table 4 presents the distance matrix for the DC and facilities.

Table 4 - The average distance among all facilities, including DC in Km

	DC	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
DC	0,00	5,73	9,50	12,35	15,40	3,37	3,60	9,23	7,13	5,67	4,13
F1	5,73	0,00	4,05	7,16	4,06	5,70	3,70	9,60	6,50	5,13	5,43
F2	9,50	4,05	0,00	5,30	3,40	5,70	7,50	14,33	9,16	8,26	5,25
F3	12,35	7,16	5,30	0,00	7,60	11,16	11,30	17,86	13,93	12,60	10,53
F4	15,40	4,06	3,40	7,60	0,00	3,56	5,26	12,23	8,76	7,33	3,05
F5	3,37	5,70	5,70	11,16	3,56	0,00	3,60	10,80	7,53	6,43	1,80
F6	3,60	3,70	7,50	11,30	5,26	3,60	0,00	6,53	4,63	4,63	3,96
F7	9,23	9,60	14,33	17,86	12,23	10,80	6,53	0,00	3,65	3,90	10,10
F8	7,13	6,50	9,16	13,93	8,76	7,53	4,63	3,65	0,00	1,26	8,16
F9	5,67	5,13	8,26	12,60	7,33	6,43	4,63	3,90	1,26	0,00	5,93
F10	4,13	5,43	5,25	10,53	3,05	1,80	3,96	10,10	8,16	5,93	0,00

Source: From Google Maps

The main objective of this work is to answer the following three questions:

- What would be the lowest transportation cost (or the best route), considering all deliveries starting from HPT (DC)?
- What would be the lowest transportation cost (or the best route) considering the installation of a new DC and now HPT as a facility (F11)?

- What would be the lowest transportation cost (or the best route), considering all deliveries to the ten facilities starting from the new DC?

Three different problems (or scenarios) are presented and solved to answer these questions in this context. The first can be understood as a simple Vehicle Routing Problem (VRP) and consists of determining the best route (minimum distance) to travel between the Hospital/Distribution Centre (HPT/DC) and the remaining ten facilities.

The second scenario is a simple Facility Location Problem (FLP) plus a simple Vehicle Routing Problem (VRP). The FLP consists of placing a new DC (different from the HPT) in an optimal place so that it will be possible to supply all the other facilities, including HPT. The VRP, similarly to the first scenario, determines the best route (minimum distance) to travel between the new DC and the remaining 11 facilities (I also include the HPT as a facility). Then, I want to compare if the cost is higher or lower than in the first scenario.

Finally, the last scenario is different from the second one. Considering the new DC, the HPT would only transfer the necessary materials to supply the ten facilities in the network. The latest DC would be responsible for handling and delivering these materials. So, I want to determine the best route (minimum distance) to travel between the new DC and the remaining ten facilities. Then, I want to compare if the cost is higher or lower than in the first and second scenarios. Of course, I am not considering the costs of installing and maintaining this new DC in both scenarios, but only the travelled distances with the new configuration.

2. The Vehicle Routing Problem (VRP)

The vehicle routing problem (VRP) is a combinatorial optimisation problem that involves finding the optimal routes for a fleet of vehicles to serve the demands of a set of customers (Asghari & MirzapourAl-e-hashem, 2020). There are several different ways of addressing the VRP in literature, such as:

- Branch and bound (Fischetti, Toth, & Vigo, 1994) – Exact Approach.
- Saving: Clark and Wright Algorithm (Pichpibul & Kawtummachai, 2013) – Heuristics (Constructive Methods).
- Multi-Route Improvement Heuristics (Kindervater & Savelsbergh, 2003; Thompson & Psaraftis, 1993) – Heuristics (Constructive Methods).
- Cluster-First, Rout-Second Algorithms (Fisher & Jaikumar, 1981; Ryan, Hjorring, & Glover, 1993) – Heuristics (2-Phase Algorithms[1])
- Constraint Programming (Rousseau, Gendreau, Pesant, & Focacci, 2004; Shaw, 1998) – Metaheurisctics.
- Simulated Annealing (Arbelaitz, Rodriguez, & Zamakola, 2001; Czech & Czarnas, 2003) – Metaheuristics.

[1] The problem is decomposed into its two natural components: (1) clustering of vertices into feasible routes and (2) actual route construction, with possible feedback loops between the two stages.

Due to the NP-Hardness of the problem, nearly all of them are heuristics and metaheuristics. This means that no exact algorithm can be guaranteed to find optimal tours within reasonable computing time when the number of cities is large (Lenstra & Kan, 1981; Tlili, Faiz, & Krichen, 2014).

Another way to face the VRP is through a Mixed and Integer Linear Programming (MILP) approach (Anbuudayasankar, Ganesh, & Mohandas, 2008; Golden, Raghavan, & Wasil, 2008; Madankumar & Rajendran, 2019). A traditional mathematical model within this approach is presented as follows (Kilby, 2013):

$$
\begin{aligned}
&\text{minimise}: \sum_{i,j} c_{ij} \sum_{k} x_{ijk} \\
&\text{subject to} \\
&\qquad \sum_{i}\sum_{k} x_{ijk} = 1 \quad \forall j \\
&\qquad \sum_{j}\sum_{k} x_{ijk} = 1 \quad \forall i \\
&\qquad \sum_{j}\sum_{k} x_{ihk} - \sum_{j}\sum_{k} x_{hjk} = 0 \quad \forall k, h \\
&\qquad \sum_{i} q_i \sum_{j} x_{ijk} \le Q_k \quad \forall k \\
&\qquad \{x_{ijk}\} \subseteq S \\
&\qquad x_{ijk} \in \{0,1\}
\end{aligned}
\qquad \text{Model (1)}
$$

The main advantage of the MILP approach is to be able to find the optimal solution. However, some disadvantages are running only for minor problems and reformulating the entire model by adding just one more constraint (Kilby, 2013). Also, depending on the constraints (heterogeneous vehicles, simultaneous delivery and pickup, and time windows), the problem can become challenging mathematically and computationally less efficient in CPU time. Finally, depending on the assumptions, MILP models dealing with the same problem can perform differently in accuracy and computational execution time (Madankumar & Rajendran, 2019). Due to these challenges and the increasing power of personal computers, there has been engagement in developing open-source optimisation tools such as OR-Tools, VROOM, and jspirit to solve real-world logistics problems (Karkula, Duda, & Skalna, 2019).

The OR-Tools is open-source software for combinatorial optimisation, seeking to find the best solution to a problem out of an extensive set of possible solutions for various problem types, such as vehicle routing, scheduling, and bin packing. It supports several different programming languages, including Python, Java, C++, and C# (Google Developers, 2021).

Although the OR-Tools is far from the other two tools for most demand instances, it can also provide reasonable quality solutions (Karkula et al., 2019), interact with different languages, and have extensive and clear documentation that includes several examples. Due to these advantages and considering that our problem is straightforward, OR-Tools could be the most appropriate tool to overcome the problem. In the next section, I will provide the solution and details on obtaining it from OR Tools.

For all the problems, it was used the IDLE (Python 3.9 64-bit) and the CPU Intel(R) Core (TM) i7-8750H CPU @ 2.20 GHz, 2.21 GHz, with 8,00 GB of installed RAM and operational system x64 based Windows 10 Home, version 20H2.

3. The problem's solution

3.1. The Scenario 1.

As described in Section 1, the first problem involves determining the best route (minimum distance) to travel between the Hospital/Distribution Centre (HPT/DC) and the remaining ten facilities. Figure 1 shows the initial locations for Scenario 1.

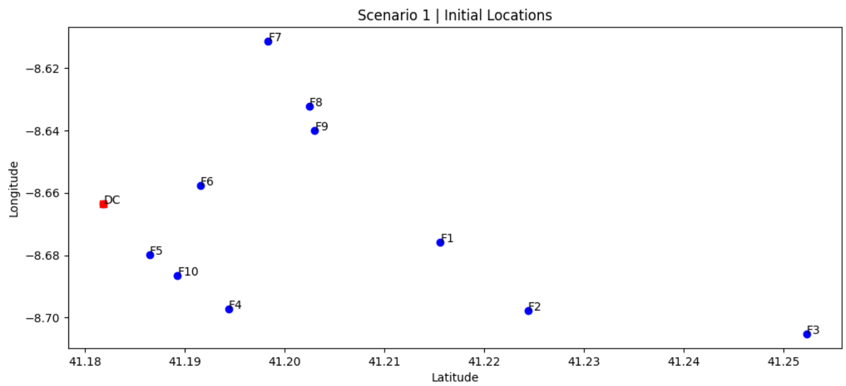


Figure 1 – Scenario 1 initial locations

The code used to solve this first scenario problem is in the file *"VRP - Scenario1.py"* attached to this report, and Figure 2 shows the solution printed in the Python console.

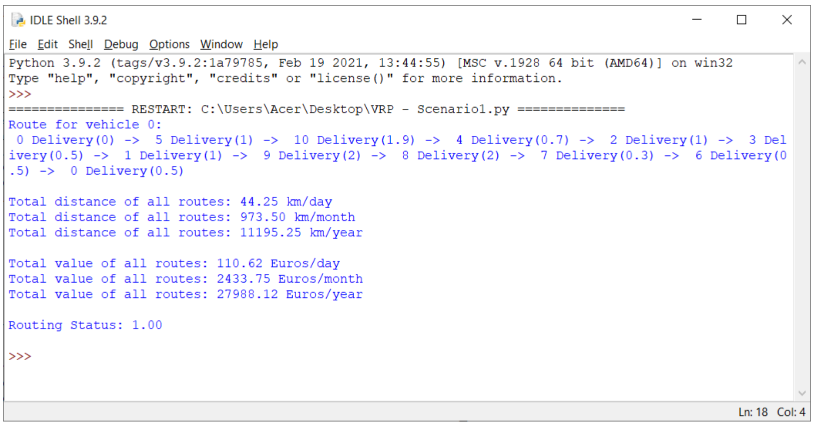


Figure 2 – Scenario 1 solution – Python console

The solution above shows the best route between the DC and the facilities and the route's distance and costs per day, month, and year. Also, it shows the routing status (see more details in the last section). Figure 3 shows the complete best route for *Scenario 1*.

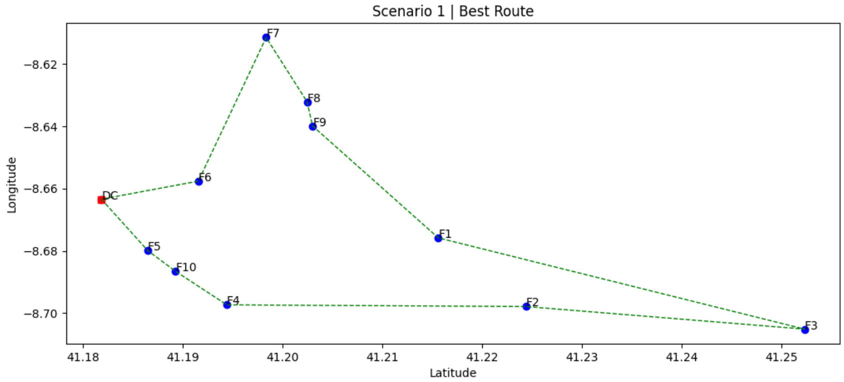


Figure 3 – Best route for Scenario 1

3.2. The Scenario 2.

As described in Section 1, scenario 2 consists of placing a new DC (different from the HPT) in an optimal place so that from this DC, it will be possible to supply all the other facilities, including HPT (F11). Then, before solving the VRP, we should solve a Facility Location Problem (FLP).

Location problems aim to configure a company's supply chain. They are also crucial for finding the optimal location of a set of facilities that minimise the cost of satisfying the demands (Hale & Moberg, 2003). The Center of Gravity Technique (CoG) is a simple way to do this.

The CoG consists of the following steps:

1. Calculate the sum of the total demands[2] (d_i) as D.
2. Calculate the sum of the weighted coordinates x_i and y_i, respectively, as X and Y:
    a. $X = \sum_{i=0}^{n} x_i * d_i , i = 1, \dots , n$
    b. $Y = \sum_{i=0}^{n} y_i * d_i , i = 1, \dots , n$
3. Obtain the new coordinates X_{New} and Y_{New}, by dividing X/D and Y/D.

The Python code for this step is the file *"FLP - Scenarios 2&3.py"*, and Figure 4 shows the solution printed in the Python console.

[2] In this problem, due to the very small and uncertain value of the demands, the number of registred patients in each facility was used as demand to calculate the weighted value of the coordinates.

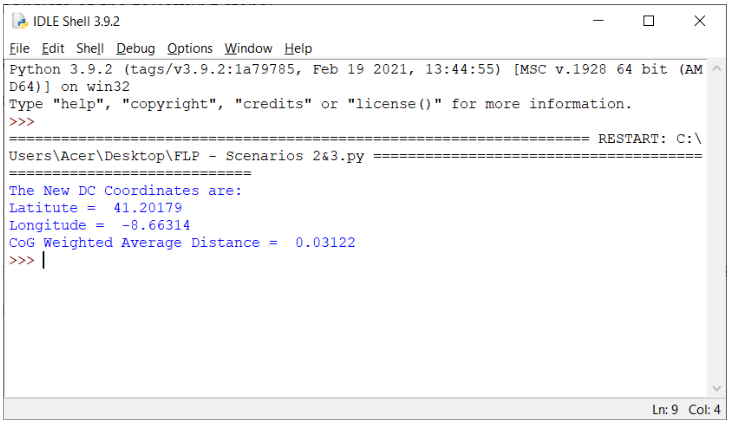


Figure 4 - Facility Location Problem solution - Python console

Since the coordinates of the new DC were obtained, Figure 5 shows the initial locations for Scenario 2.

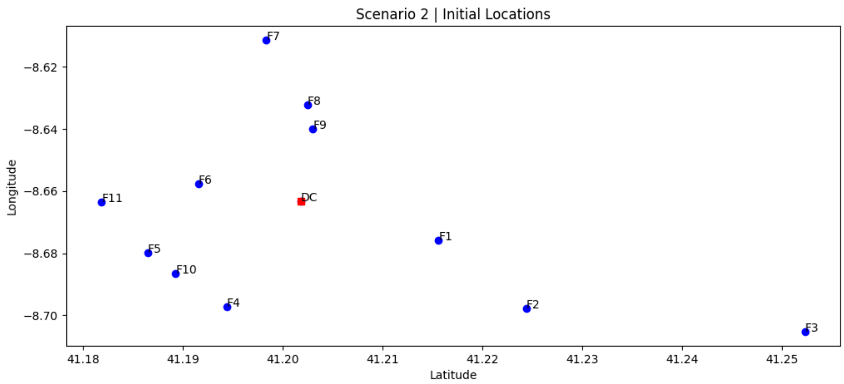
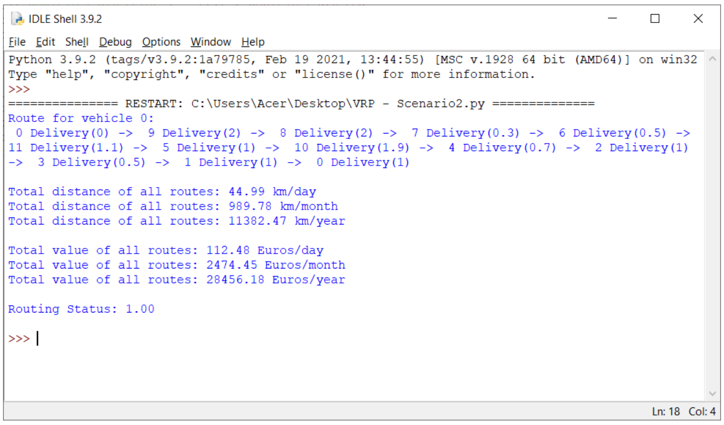


Figure 5 - Scenario 2 initial locations

Afterward, the following changes were made in the code used to solve the *"Scenario 1"* problem while the other parts remained the same.

1. The old DC (HPT) is now a facility (F11) and should be supplied by the new DC. So, the distance matrix should be modified to accommodate this new configuration.
2. Since F11 is a new facility to be supplied, a demand of 1.1 was estimated and included appropriately in the code.

The code used to solve this first scenario problem is in the file *"VRP – Scenario2.py"* attached to this report, and Figure 6 shows the solution printed in the Python console.



```
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\Acer\Desktop\VRP - Scenario2.py =============
Route for vehicle 0:
 0 Delivery(0) -> 9 Delivery(2) -> 8 Delivery(2) -> 7 Delivery(0.3) -> 6 Delivery(0.5) ->
11 Delivery(1.1) -> 5 Delivery(1) -> 10 Delivery(1.9) -> 4 Delivery(0.7) -> 2 Delivery(1)
-> 3 Delivery(0.5) -> 1 Delivery(1) -> 0 Delivery(1)

Total distance of all routes: 44.99 km/day
Total distance of all routes: 989.78 km/month
Total distance of all routes: 11382.47 km/year

Total value of all routes: 112.48 Euros/day
Total value of all routes: 2474.45 Euros/month
Total value of all routes: 28456.18 Euros/year

Routing Status: 1.00

>>> |
Ln: 18 Col: 4
```

Figure 6 - Scenario 2 solution – Python console

The solution above shows the best route between the DC and the facilities and the route's distance and costs per day, month, and year. Also, it shows the routing status (see more details in the last section). Figure 7 shows the complete best route for *Scenario 2*.

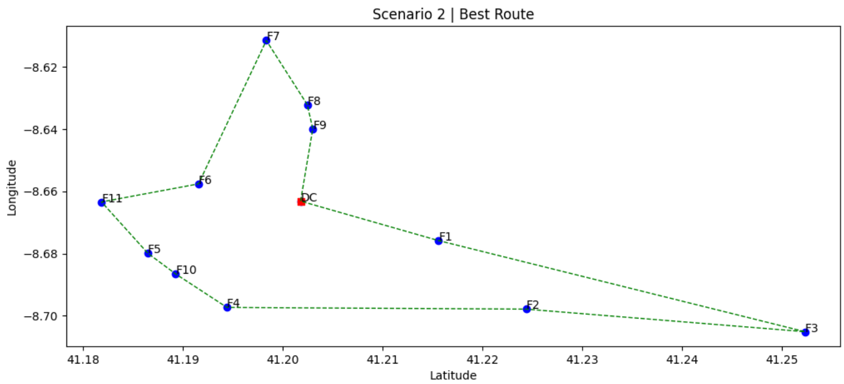


Figure 7 - Best route for Scenario 2

3.3. The Scenario 3.

As described in Section 1, in scenario 3, the HPT would only transfer to this DC the necessary materials to supply the ten facilities in the network. The new DC would be responsible for handling and delivering these materials. So, the problem is to determine the best route (minimum distance) to travel between the new DC and the remaining ten facilities. Then, the distance travelled (round trip) from the HPT to DC will be added. Figure 8 shows the initial locations for Scenario 1.

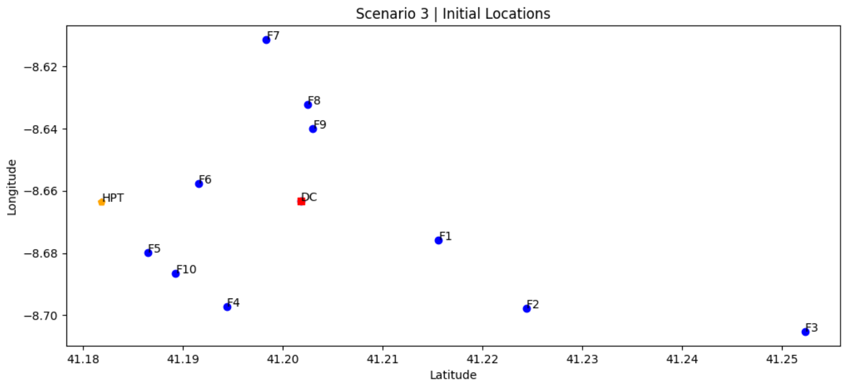


Figure 8 - Scenario 3 initial locations

Afterward, the following changes were made in the code used to solve the *"Scenario 1"* problem while the other parts remained the same.

1. Since the DC should not supply HPT, its demand should have been considered.
2. The distance matrix was modified concerning this new configuration.
3. The objective function was modified to consider the distance between HPT and DC (round trip).

The code used to solve this first scenario problem is in the file *"VRP – Scenario3.py"* attached to this report, and Figure 9 shows the solution printed in the Python console.

The solution shows the best route between the DC and the facilities, and the route's distance and costs per day, month, and year. Also, it shows the routing status (see more details in the last section). Figure 10 shows the complete best route for *Scenario 3*.

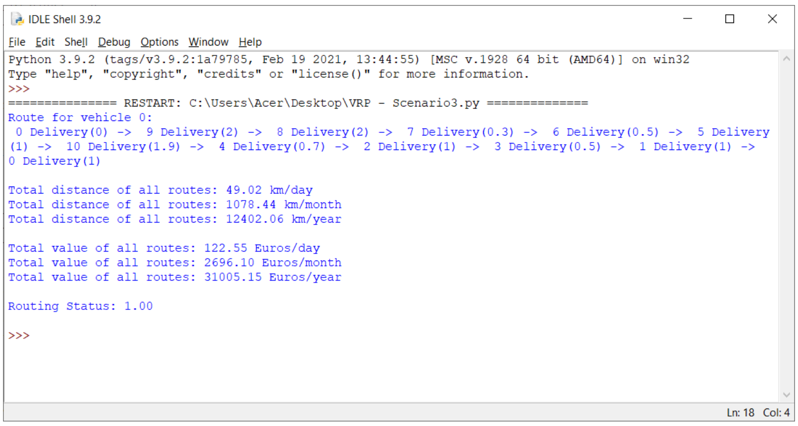


Figure 9 - Scenario 3 solution – Python console

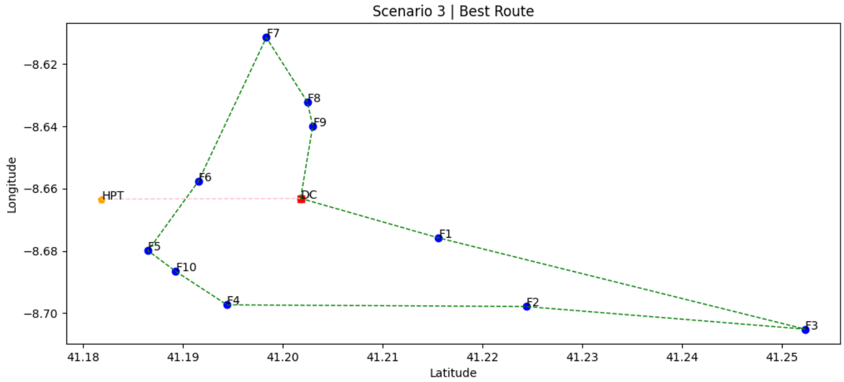


Figure 10 - Best route for Scenario 3

4. Scenario comparison

Since the three proposed scenarios were run, it is possible to compare their results in Table 5.

Table 5 - Scenario comparison

	Route (Km)			Value (€)		
	Daily	Monthly	Yearly	Daily	Monthly	Yearly
Scenario_1	44,25	973,50	11 195,25	110,62	2 433,75	27 988,12
Scenario_2	44,99	989,78	11 382,47	112,48	2 474,45	28 456,18
Scenario_3	49,02	1 078,44	12 402,06	122,55	2 696,10	31 005,15
Max_Available	**65,00**	**1 430,00**	**16 445,00**	**151,53**	**4 846,00**	**58 152,00**

Scenario 2 is slightly worse than Scenario 1, even considering installation and maintenance costs were not added to the problem.

Scenario 3 is also worse than Scenario 1, even considering installation and maintenance costs were not added to the problem. However, depending on the logistics strategy (e.g., direct delivery by suppliers in both HPT and DC without the need for HPT delivers to DC), if HLN cut the cost of HPT – DC – HPT, the total distance would be 41,62 km/day (101,97€/day). Annually, the cost to supply the ten facilities would be 25 798,41€, which is 2 189,71€, lower than in Scenario 1. Supposing the investment cost is I_{cost}, the Payback Period (PP) of this investment could be calculated as:

$$PP = \frac{I_{cost}}{2\ 189{,}71}$$

Finally, based on the table, it is possible to see that if the hospital used optimisation methods to verify the contract, the contractual availability might be overestimated annually by 46,66% (Δ = (Max_Available – Scenario_3)/Max_Available).

References

Anbuudayasankar, S. P., Ganesh, K., & Mohandas, K. (2008). Mixed-integer linear programming for vehicle routing problem with simultaneous delivery and pickup with maximum route length. *The International Journal of Applied Management and Technology*, *6*(I), 31–52.

Arbelaitz, O., Rodriguez, C., & Zamakola, I. (2001). Low-cost parallel solutions for the VRPTW optimisation problem. *Proceedings International Conference on Parallel Processing Workshops*, *2001-Janua*, 176–181. https://doi.org/10.1109/ICPPW.2001.951932

Asghari, M., & MirzapourAl-e-hashem, S. M. J. (2020). New Advances in Vehicle Routing Problems: A Literature Review to Explore the Future. In *Green Transportation and New Advances in Vehicle Routing Problems*. https://doi.org/10.1007/978-3-030-45312-1_1

Czech, Z. J., & Czarnas, P. (2003). Parallel simulated annealing for the vehicle routing problem with time windows. *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-Based Processing (EUROMICRO-PDP' 02)*, 376–383. https://doi.org/10.1109/empdp.2002.994313

Fischetti, M., Toth, P., & Vigo, D. (1994). A Branch-and-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs. *Operations Research*, *42*(5), 846–859. https://doi.org/10.1287/opre.42.5.846

Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, *11*(2), 109–124. https://doi.org/10.1002/net.3230110205

Golden, B., Raghavan, S., & Wasil, E. (2008). The Vehicle Routing Problem: Latest Advances and New Challenges. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *Operations Research/ Computer Science Interfaces Series*. https://doi.org/10.1007/978-0-387-77778-8

Google Developers. (2021). OR-Tools | Google Developers. Retrieved May 2, 2021, from Google website: https://developers.google.com/optimization

Hale, T. S., & Moberg, C. R. (2003). Location Science Research: A Review. *Annals of Operations Research*, *123*(1–4), 21–35. https://doi.org/10.1023/A:1026110926707

Karkula, M., Duda, J., & Skalna, I. (2019). Comparison of capabilities of recent open-source tools for solving capacitated vehicle routing problems with time windows. *Proceedings of the Carpathian Logistics Congress - CLC'2019*, 72–77. Retrieved from https://www.confer.cz/clc/2019/2922-comparison-of-capabilities-of-recent-open-source-tools-for-solving-capacitated-vehicle-routing-problems

Kilby, P. (2013). The Vehicle Routing Problem, Using Constraint Programming in the Real World. In A. for C. P.- ACP (Ed.), *19th International Conference on Principles and Practice of Constraint Programming - CP'2013*. Uppsala, Sweden.

Kindervater, G. A. P., & Savelsbergh, M. W. P. (2003). Vehicle routing: handling edge exchanges. In *Local Search in Combinatorial Optimization* (pp. 337–360). https://doi.org/10.1515/9780691187563-013

Lenstra, J. K., & Kan, A. H. G. R. (1981). Complexity of vehicle routing and scheduling problems.

Networks, *11*(2), 221–227. https://doi.org/10.1002/net.3230110211

Madankumar, S., & Rajendran, C. (2019). A mixed integer linear programming model for the vehicle routing problem with simultaneous delivery and pickup by heterogeneous vehicles, and constrained by time windows. *Sādhanā*, *44*(2), 39. https://doi.org/10.1007/s12046-018-1048-y

Pichpibul, T., & Kawtummachai, R. (2013). A Heuristic Approach Based on Clarke-Wright Algorithm for Open Vehicle Routing Problem. *The Scientific World Journal*, *2013*, 1–11. https://doi.org/10.1155/2013/874349

Rousseau, L.-M., Gendreau, M., Pesant, G., & Focacci, F. (2004). Solving VRPTWs with Constraint Programming Based Column Generation. *Annals of Operations Research*, *130*(1–4), 199–216. https://doi.org/10.1023/B:ANOR.0000032576.73681.29

Ryan, D. M., Hjorring, C., & Glover, F. (1993). Extensions of the Petal Method for Vehicle Routeing. *Journal of the Operational Research Society*, *44*(3), 289–296. https://doi.org/10.1038/sj/jors/0440309

Shaw, P. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 1520, pp. 417–431). https://doi.org/10.1007/3-540-49481-2_30

Thompson, P. M., & Psaraftis, H. N. (1993). Cyclic Transfer Algorithm for Multivehicle Routing and Scheduling Problems. *Operations Research*, *41*(5), 935–946. https://doi.org/10.1287/opre.41.5.935

Tlili, T., Faiz, S., & Krichen, S. (2014). A Hybrid Metaheuristic for the Distance-constrained Capacitated Vehicle Routing Problem. *Procedia - Social and Behavioral Sciences*, *109*, 779–783. https://doi.org/10.1016/j.sbspro.2013.12.543