

---

*Research article*

## A 3D proposal for the visualization of speed in railway networks

Alberto Almech<sup>1</sup> and Eugenio Roanes-Lozano<sup>2,\*</sup>

<sup>1</sup> Universidad Complutense de Madrid, Madrid, Spain

<sup>2</sup> Instituto de Matemática Interdisciplinar & Departamento de Didáctica de las Ciencias Experimentales, Sociales y Matemáticas, Facultad de Educación, Universidad Complutense de Madrid, Spain

\* **Correspondence:** Email: [eroanes@ucm.es](mailto:eroanes@ucm.es); Tel: +34913946248; Fax: +34913946116.

**Abstract:** This article deals with a proposal for visualizing the speed of the different sections of the lines of a railway network that has been implemented in the computer algebra system Maple. The idea is to organize the data (the speed in the different sections of the railway network) as a weighted graph. The endpoints of the sections considered are the vertices of the graph and the edges are the sections of the railway lines of the network. The weights of the edges reflect the different speeds in the sections of the network. The vertices of the graph are drawn in the  $xy$  plane according to their geographical coordinates. The edges are represented by segments in the  $xy$  plane. Vertical rectangles are lifted from these segments according to the weights of the edges (as a kind of wrinkled histogram). Two different methods are proposed to compute the height of the rectangles: one directly considers the difference of speeds with respect to the maximum average speed and the other calculates the height so that the area of the rectangle represents the time required to traverse the section. This way the speeds of the different sections of the lines can be easily visualized (in 3D). The underlying mathematics is elementary, but the implementation is complex and makes extensive use of the possibilities of Maple's plot package.

**Keywords:** graph theory; railway networks; visualization; real world applications; computer algebra systems

**Mathematics Subject Classification:** 00A66, 05C62, 05C90, 90C35

---

### 1. Introduction

Passengers' decision making when traveling is more related to timings than to real distances.

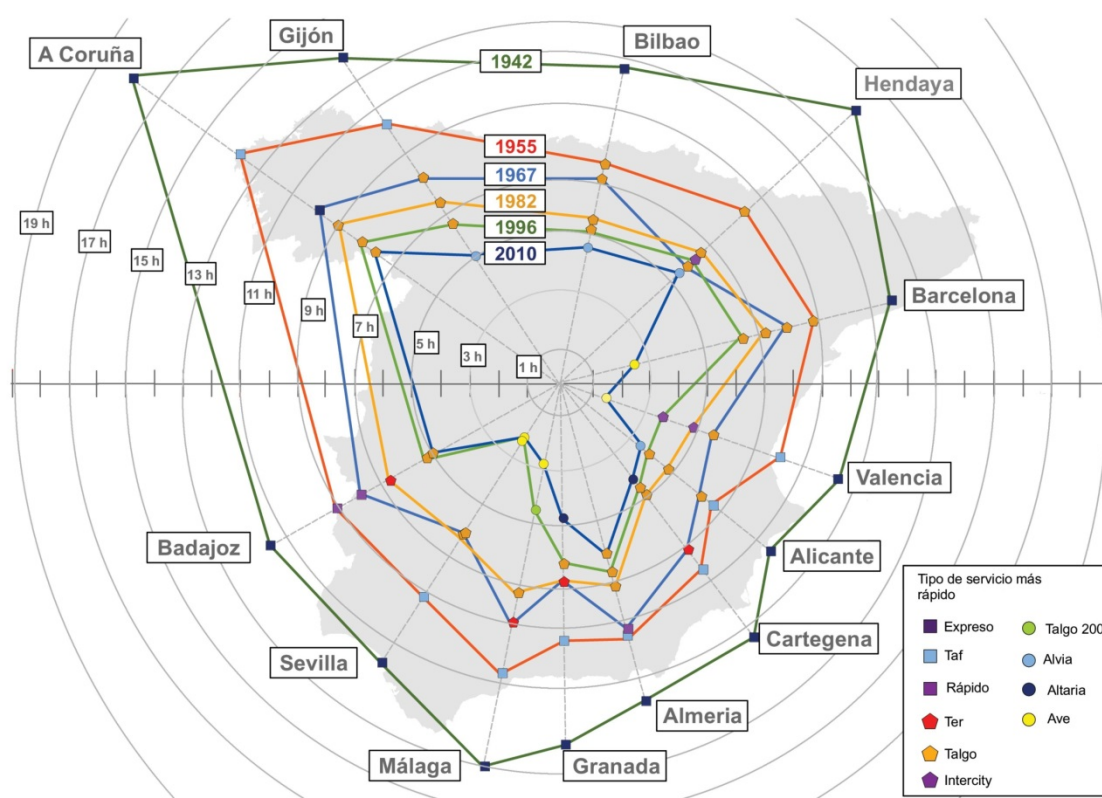
For instance, a 2 hours flight between European capitals is nowadays considered “a short travel” (it is considered “normal” to attend a business lunch and return the same day).

This idea has been transferred to railway travels, especially after the arrival of high-speed trains. For instance, the Madrid-Barcelona high-speed line was designed and the rolling stock was selected with a timing goal in mind: that it could compete with the airlines operating this route.

We are very concerned about engineering data visualization. We have previously worked on visualization of radial railway networks, mixing the ideas underlying anamorphosis maps and polar area diagrams. We denoted our graphs isochrone circle graphs [1,2].

In polar area diagrams (diagrams similar to usual pie charts), sectors have equal angles and their radii are adjusted so that the area of the sector corresponds to the value of the variable. They were introduced by Florence Nightingale back in the XIXth century [3].

Meanwhile, in anamorphosis maps (also known as central point cartograms and distance cartograms) [4–6], the geographical heading from the central origin is respected but the distances from the latter point depend on the travel times (Figure 1). They were also introduced in the XIXth century [7,8] and are now relatively common, even in non-scientific sources [9,10]. Related anamorphosis processes have been applied in other different ways to visually show data (see, for instance, [11]).

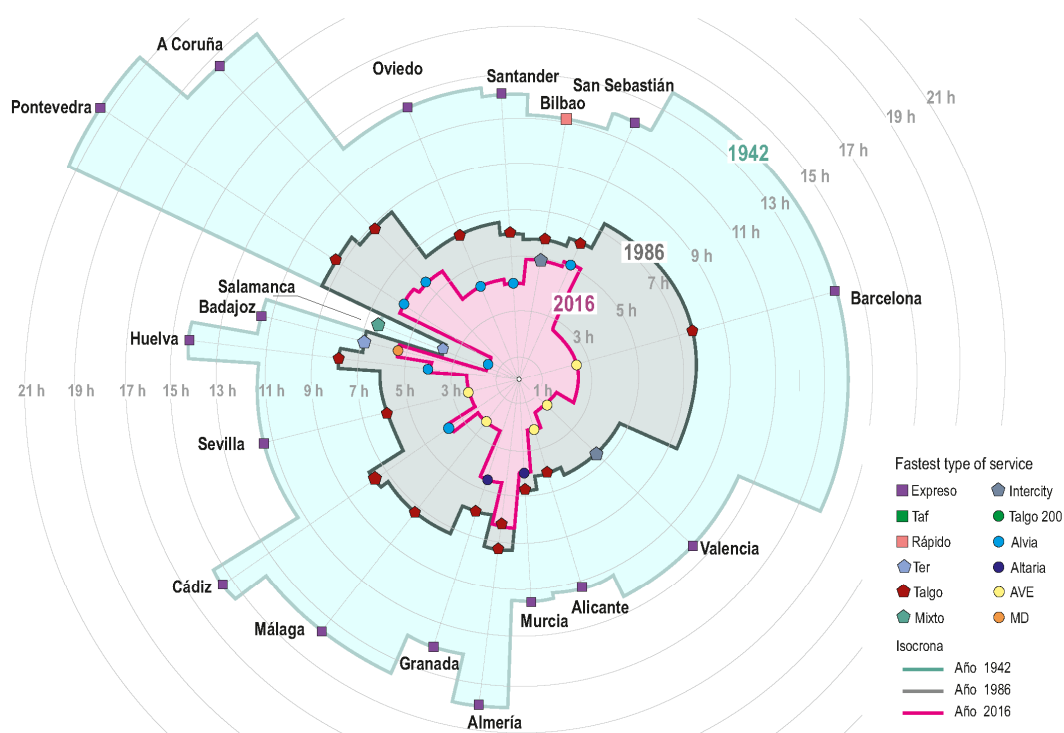


**Figure 1.** Anamorphosis map of the Spanish railway network (1942–2010). Courtesy of the *Fundación de los Ferrocarriles Españoles* (*Spanish Railway Foundation*). The English translation of “Tipo de servicio más rápido” is “Fastest type of service”.

In our isochrone circle graphs, the heading of a certain peripheral city from the center of the graph is chosen as the starting direction. The rest of the peripheral destinations are allocated clockwise or counterclockwise in headings and distances such that the angle of the sectors depend on the population served and the radii depend on the time to the destination (instead of on the distance to the destination). Our idea is now adopted, for instance, by the *Fundación de los Ferrocarriles Españoles* (Spanish Railway Foundation) [12] in their newest diagrams (Figure 2).

There are proposals for alternative deformations of the map for visualizing speed in railway networks [13]. There are also 2D proposals for visualization of subway networks that use the thickness of the lines [14].

Another possibility is the use of accelerated time simulations to generate visual information [15–18]. Obviously, visualization of data can be successfully applied not only to transportation engineering, but also to mathematics [19,20], sciences [21], etc.



**Figure 2.** Three superimposed isochrone circle maps (1942, 1986, 2016) of the Spanish railway network. Courtesy of the *Fundación de los Ferrocarriles Españoles* (Spanish Railway Foundation).

Graphs are very powerful tools for modeling and visualization. We can distinguish [22,23]:

- A *graph*,  $G$ , is an ordered pair  $G = (V, E)$ , where  $V$  is a set of elements called *vertices* and  $E$  is a set of sets of the form  $\{v1, v2\}$ , where  $v1, v2 \in V$ . The elements of  $E$  are called *edges*.
- A *directed graph* (also called *digraph*),  $G$ , is an ordered pair  $G = (V, E)$ , where  $V$  is a set of

elements called *vertices* and  $E$  is a set of ordered pairs of the form  $(v1, v2)$ , where:  $v1, v2 \in V$  and  $v1 \neq v2$ . The elements of  $E$  are called *directed edges* or *arcs*.

- A *weighted graph* is a graph in which a weight (a number) is assigned to each edge.

Graphs have applications in many different fields of engineering, such as the design and analysis of electric circuits [24], origami structures [25], space structures [26], railway interlocking systems [27], knowledge engineering [28], etc.

The new idea presented in this paper is the following:

- Firstly, a weighted graph in which the edges are sections of the railway lines of the network (and, therefore, the endpoints of the sections considered are the vertices of the graph) is considered. A weight is assigned to each edge. In this case, each weight represents a datum related to the average speed in that section.
- Secondly, a 3D representation of the weighted graph, that uses vertical rectangles, is computed and represented, creating a kind of relief map that presents the average speed in the different sections in a very visual way.

Like if we were following the summit line of a chain of mountains, we can follow the upper horizontal sides of the rectangles corresponding to the sections of a railway line in the 3D representation of the network. The highest ones correspond to the sections with the lowest average speed. If more than one railway line is represented, it looks like drainage divides.

The height of the rectangles can be assigned in two ways in this proposal:

- it can be the difference of average speeds between that section and the sections with the highest average speed in the network (then the rectangles of height 0, that is, the rectangles that have collapsed to a segment in plane  $z=0$ , correspond to the sections with the highest average speed in the network),

or

- it can be a value such that the area of the rectangle is the time required to traverse that section of the network (this way, the sum of the areas of the rectangles corresponding to a line is the total travel time, similarly to how the area under a curve is the definite integral of the corresponding function).

The package is implemented in the computer algebra system Maple 2020<sup>1</sup> and it takes advantage of the powerful command `poligonplot3d` of Maple's `plots` package (among other commands of this package) and of `ListTools` package. A poster on a first version of this work was presented at FuzzyMAD 2019 conference [29].

The article is illustrated with graphs produced from real data obtained from Renfe Operadora<sup>2</sup> timings in SW Spain lines. This part of Spain includes the oldest high-speed line (1992), as well as other high-speed and classic lines.

The underlying mathematics is simple, as happens in papers [1,2], but we believe it is an interesting 3D visualization that makes clear which regions are best and worst served by the railway network and which parts of a line are responsible for a low total travel time. Visually introducing

<sup>1</sup> Maple is a trademark of Waterloo Maple Inc.

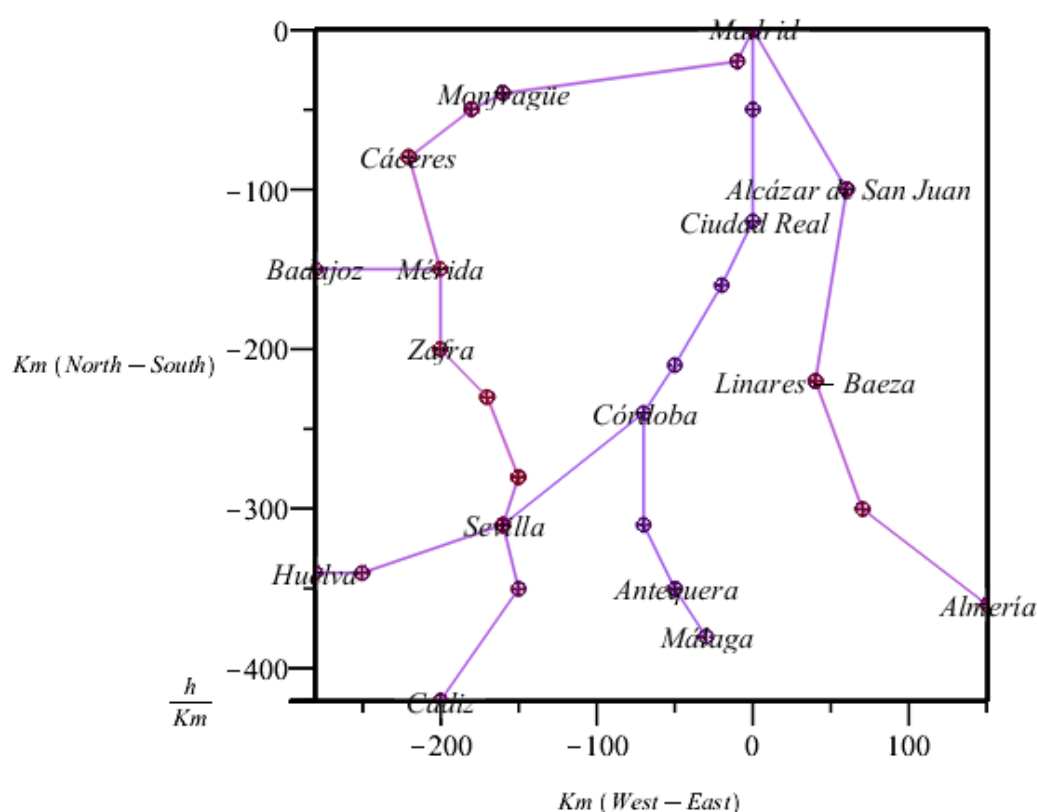
<sup>2</sup> Renfe Operadora is a public Spanish entity that provides passenger and freight railway transport services.

data is important, and can be the key when transmitting information to average citizens. We know of no similar work. Let us finally note that introducing ideas or techniques from other fields in transportation engineering can be very fruitful [30].

## 2. Materials and method

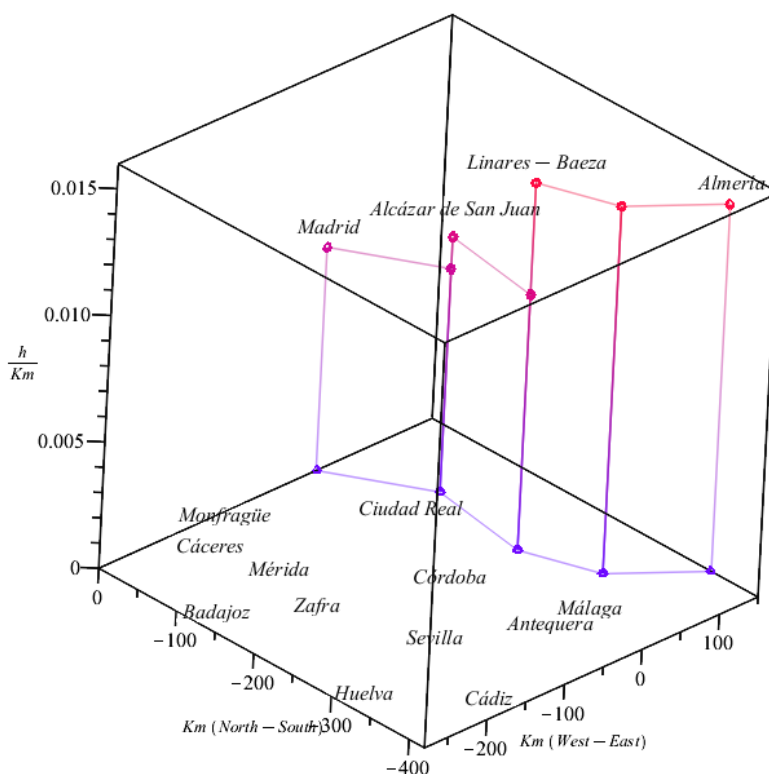
### 2.1. Formulation

The weighted graph is initially represented in the  $xy$  plane (without considering the weights of the edges). The vertices are allocated according to their corresponding coordinates. Segments are used to represent the sections of the lines considered (the edges of the graph –without weights). This way a simplified map of the railway network is obtained (Figure 3).

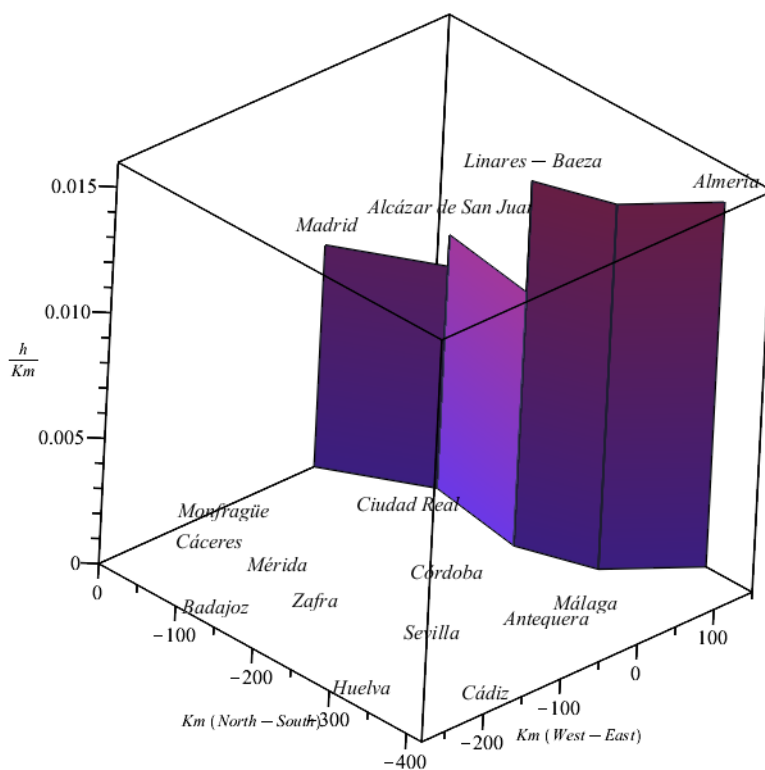


**Figure 3.** A simplified map of the railway subnetwork considered as illustration in this paper (from SW Spain). Some are standard track gauge and others are Iberian track gauge.

Afterwards vertical rectangles are built according to the weight of each edge (Figure 4), like in a histogram where the rectangles are not coplanar but form different angles with the adjacent ones instead. It is visually clear that the third and fourth sections disadvantage the total timing in the Madrid–Almería train timetable represented in Figures 4 and 5 (note that only the best timetables for each route are considered).



**Figure 4.** A frame proposal of 3D representation of the timetable of the Talgo train Madrid–Almería (that has three intermediate stops).



**Figure 5.** A rectangles proposal of the representation of the timetable of the train of Figure 4.

## 2.2. Method I: Inverse of average speeds

For each vertical rectangle, the height assigned is

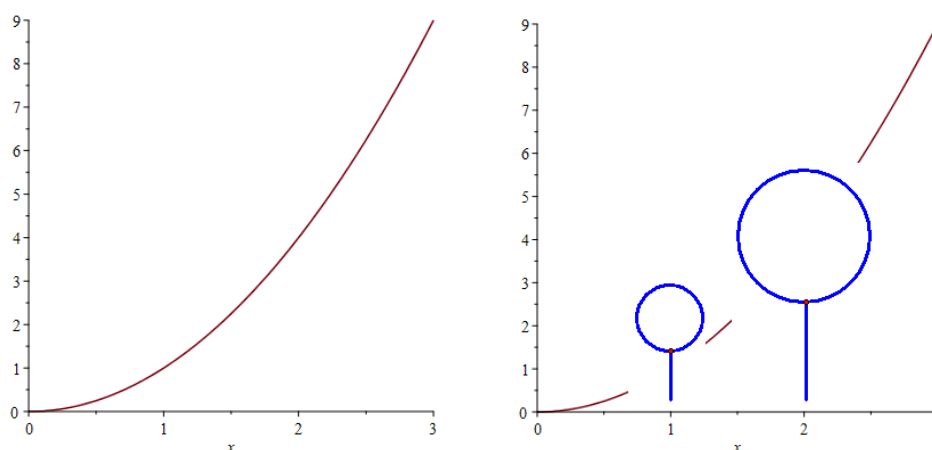
$$h = \Delta v^{-1} = \Delta t / \Delta s \quad (1)$$

where  $\Delta s$  is the length of the section and  $\Delta t$  the time to traverse it. The reason for this choice is that, this way, the area of the rectangle is:

$$(\Delta t / \Delta s) \Delta s = \Delta t \quad (2)$$

(that is, the area of each rectangle is the time required to traverse that section of the line).

This is an important issue because, many times, a drawing representing, for instance, the production of a company, is incorporated to the plot of the function “production level”. At another point of the function, for instance, when the production is doubled, the same drawing is included, dilated by a 2:1 ratio (Figure 6). This is not fair, as the area of the second drawing is four times the area of the first one, visually introducing the underlying idea that the production has increased much more.



**Figure 6.** Two plots representing the evolution of the production of a lollipop candies factory. On the left the real curve, on the right the curve “illustrated” with simplified drawings of two lollipop candies. The bigger drawing is the image of the smaller one in a homothety of ratio 2. This gives the impression that the production has quadrupled instead of doubled from the first year to the second year.

## 2.3. Method II: Differences of average speeds

Let  $v_0$  be the maximum speed in the sections of the network. For each vertical rectangle, the height assigned is

$$h = v - v_0 = (\Delta s / \Delta t) - v_0 \quad (3)$$

The reason for this choice is that directly representing speeds in the vertical axis is more common. Moreover, it gives a direct measure of the “height” of the “speed jumps” from one section of the network to an adjacent one.

### 3. Results

Let us show how to obtain the different plots (the complete code is included in the Appendix).

#### 3.1. Real example, inverse of average speeds (Method I)

Let us show how Figures 3, 4 and 5 were generated. As said in the Introduction, two Maple standard packages and the new package have to be firstly loaded:

```
> restart;
> with(plots):
> with(ListTools):
> load(C:/.../rectangles.mpl);
```

(where the ... represent the path to the file `rectangles.mpl` containing the code of the new package).

Afterwards the initial data of the representation have to be introduced, for instance:

```
> partition := 10:
> origin     := [40,45]:
> abscissa   := 60:
> ordinate   := 45:
```

The data chosen mean: gap between vertices of the lattice: 10 km; coordinates of the origin of the lattice (Madrid): (40,45); number of vertices in the abscissa axis direction: 60; number of vertices in the ordinate axis direction: 45 (therefore the lattice represent an area of  $600 \text{ km} \times 450 \text{ km}$ ).

The package can handle the existence of two track gauges. In this case the train timetables have been stored in two variables, denoted `AV`<sup>3</sup> (“standard gauge”) and `IB` (“Iberian gauge”), respectively. Each route (denoted `trail`) is introduced section by section, alternating the details of the vertices and the weighted edges between them, in the following way:

```
[abscissa, ordinate, name, label yes/no] ,
inverse of the average speed in the section
```

and `AV` and `IB` are updated using `Introtrail` procedure. A detailed example is given afterwards.

The best high-speed trains (AVE) in the Madrid–Seville route (non-stop, 2 h 21 m) and Córdoba–Málaga (1 intermediate stop, 55 m) can be introduced as follows<sup>4</sup> (in this case they are provisionally allocated in variable `trail` before updating the trains’ data):

```
> AV := [];
> trail := [[0, 0, 'Madrid', 1], 141/471, [0, -5, '`La Sagra`', 0], 141/471,
[0, -12, '`Ciudad Real`', 1], 141/471, [-2, -16, 'Puertollano', 0],
141/471, [-5, -21, 'Villanueva', 0], 141/471, [-7, -24, 'Córdoba', 1],
141/471, [-16, -31, 'Sevilla', 1]];
```

<sup>3</sup> The Spanish high-speed railway network is standard gauge (1,435 mm) while the Spanish classic railway network is Iberian gauge (1,668 mm). The translation to Spanish of “high-speed” is “alta velocidad”.

<sup>4</sup> Note that the inverses of the speeds are all equal for the first train: that is because this is a non-stop service.



---

```
> Introtrail(trail, 'av');
> trail := [[-7, -24, 'Córdoba', 1], 31/111, [-7, -31, 'Puenete Genil',
0], 31/111, [-5, -35, 'Antequera', 1], 24/58, [-3, -38, 'Málaga', 1]];
> Introtrail(trail, 'av');
```

Let us remark that not all stops have been introduced in this example for the sake of clarity (for instance the R. Exprés below has very frequent stops). On the other hand, some stations without train stop have been introduced in order the plot to be more accurate to the real map (like Puenete Genil in the Córdoba–Sevilla timetable above). Of course the level of detail can be chosen by the user. For instance, all stops can be introduced in a regional route if analyzing just that route.

Some Iberian gauge train timetables are similarly introduced below: MD<sup>5</sup> Madrid–Badajoz (5 h 18 m, 11 intermediate stops), Talgo<sup>6</sup> Madrid–Almería (6 h 28 m, 3 intermediate stops), Alvia<sup>7</sup> Sevilla–Cádiz (1 h 32 m, 3 intermediate stops), Alvia Sevilla–Huelva (1 h 22 m, 1 intermediate stop), R. Exprés<sup>8</sup> Sevilla–Mérida (3 h 34 m, 14 intermediate stops).

```
IB := [];
trail := [[0, 0, 'Madrid', 1], 150/234, [-1, -2, 'Humanes', 0], 150/234,
[-16, -4, 'Monfrague', 1], 9/14, [-18, -5, 'Mirabel', 1], 50/66, [-22,
-8, 'Cáceres', 1], 71/72, [-20, -15, 'Mérida', 1], 38/59, [-28, -15,
'Badajoz', 1]];
Introtrail(trail, 'ib');
trail := [[0, 0, 'Madrid', 1], 78/147, [6, -10, 'Alcázar de San Juan',
1], 100/165, [4, -22, 'Linares - Baeza', 1], 210/241, [7, -30, 'Moreda',
0], 210/241, [15, -36, 'Almería', 1]];
Introtrail(trail, 'ib');
trail := [[-16, -31, 'Sevilla', 1], 92/153, [-15, -35, 'Utrera', 0],
92/153, [-20, -42, 'Cádiz', 1]];
Introtrail(trail, 'ib');
trail := [[-16, -31, 'Sevilla', 1], 82/119, [-25, -34, 'Y', 0], 82/119,
[-28, -34, 'Huelva', 1]];
Introtrail(trail, 'ib');
trail := [[-16, -31, 'Sevilla', 1], 21/27, [-15, -28, 'Los Rosales',
0], 106/99, [-17, -23, 'Llerena', 0], 39/40, [-20, -20, 'Zafra', 1],
48/65, [-20, -15, 'Mérida', 1]];
Introtrail(trail, 'ib');
```

---

<sup>5</sup> MD is the acronym for “Media Distancia” (Middle Distance) services.

<sup>6</sup> Talgo locomotive hauled trains have lightweight, independent wheels, natural tilting coaches, offering high speed and quality services in conventional lines. Renfe also owns Talgo trains operated on high-speed lines (classes 102 and 112) as well as dual gauge trains (classes 130 and 730, the latter hybrid).

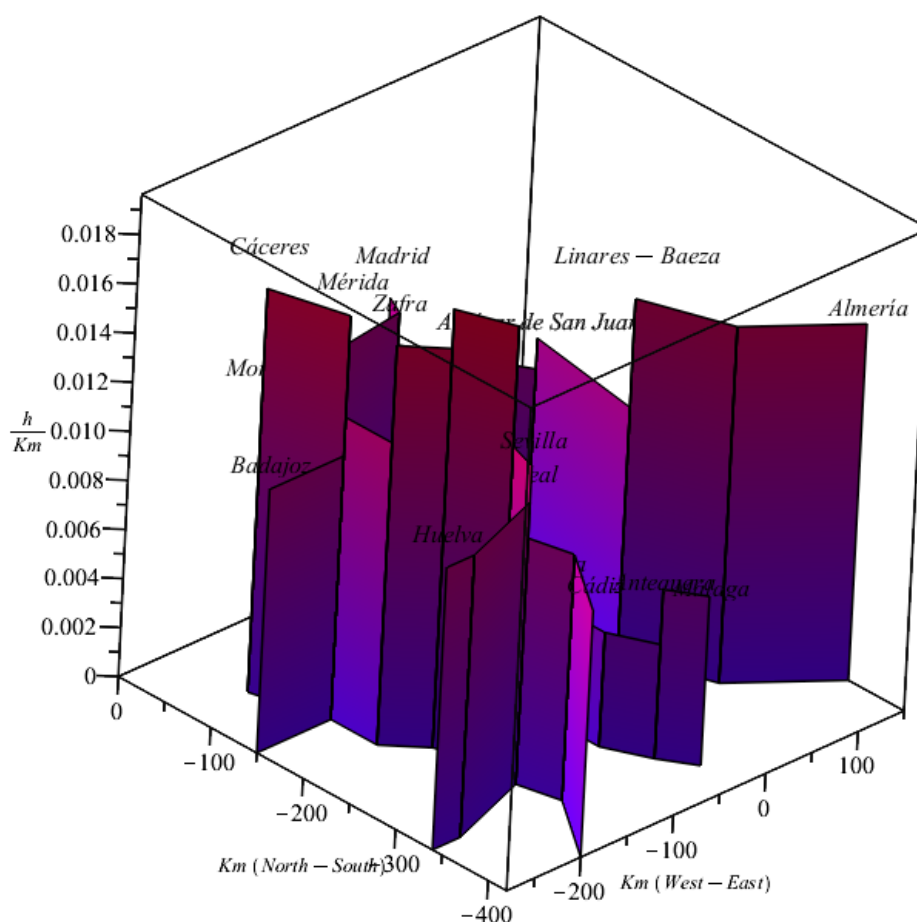
<sup>7</sup> Alvia is a commercial name for high speed and quality services in routes along conventional and high speed lines (the rolling stock used is dual gauge).

<sup>8</sup> Regional Exprés is the commercial name for low speed, multiple stops regional services (sometimes it is the only service available in low traffic lines).

The timetables of both track gauges can then be joined in a list and plotted altogether (in this case they are allocated in variable ``3Dtrains``):

```
`3Dtrains` := GraphicR([op(AV), op(IB)]);
PlotR(`3Dtrains`, [-130, 60, 5], 'surfacewireframe', 'A');
```

The plot generated can be found in Figure 7. Note that the 'surfacewireframe' option corresponds to a plot with the interior of the rectangles filled. The three numbers in the second option represent the orientation of the plot (see Maple's help `?plot3d/options` for details).



**Figure 7.** Representation of the speeds of all trains considered in the example along their corresponding routes (Method I).

It is also possible to plot only some of the trains introduced. For instance, if we only want to plot the rectangles corresponding to the second train of list IB, that is `IB[2]`, and no train of the first list, it is enough to type:

```
`3Dtrains` := GraphicR([], IB[2]);
PlotR(`3Dtrains`, [-130, 60, 5], 'surfacewireframe', 'A');
```

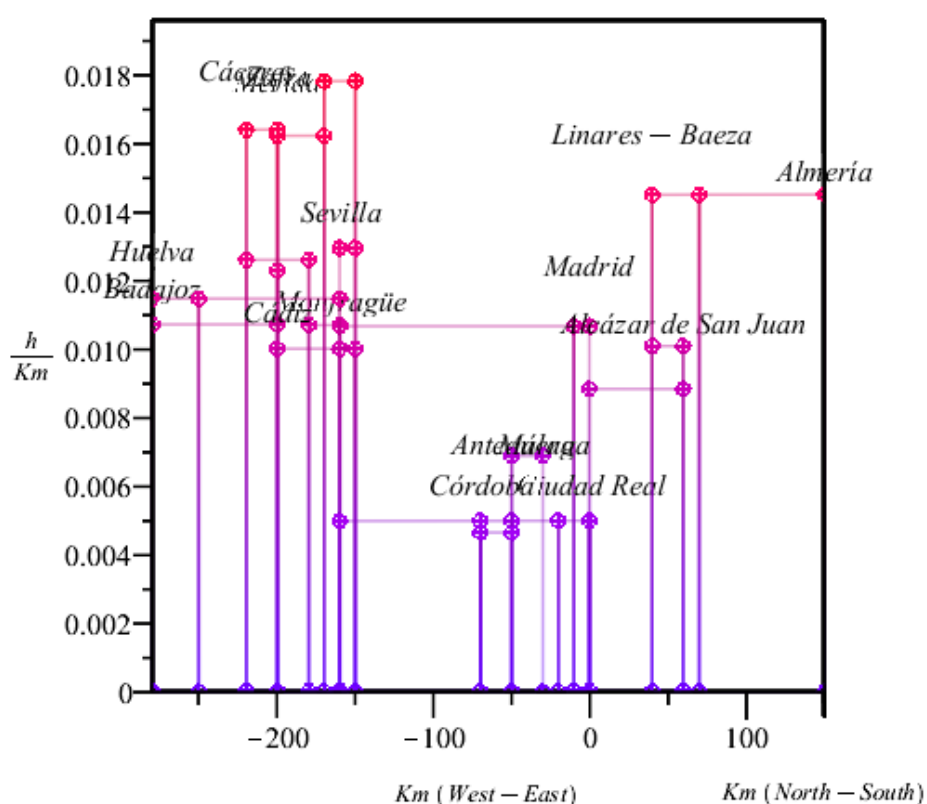
and the plot of Figure 5 is obtained. If the rectangles are not to be filled, the option 'pointline' has to be typed instead of 'surfacewireframe'. This way, the plot of Figure 4 is obtained typing:

```
`3Dtrains` := GraphicR([], IB[2]);
PlotR(`3Dtrains`, [-130, 60, 5], 'pointline', 'A');
```

Altering the orientation of the plot can produce other interesting views. For instance, the 2D map of the network considered (Figure 3) can be obtained from a zenithal view of the 3D plot by typing:

```
`3Dtrains` := GraphicR([op(AV), op(IB)]);
PlotR(`3Dtrains`, [-90, 0, 0], 'pointline', 'A');
```

Similarly, a side view of the frames using the 'pointline' option, provides another comparative vision of the average speed in the different sections (Figure 8), where, for instance, the lowest and highest speed sections are distinguishable.



**Figure 8.** Side view of the frame 3D plot (Method I).

### 3.2. Real example, increase of average speeds(Method II)

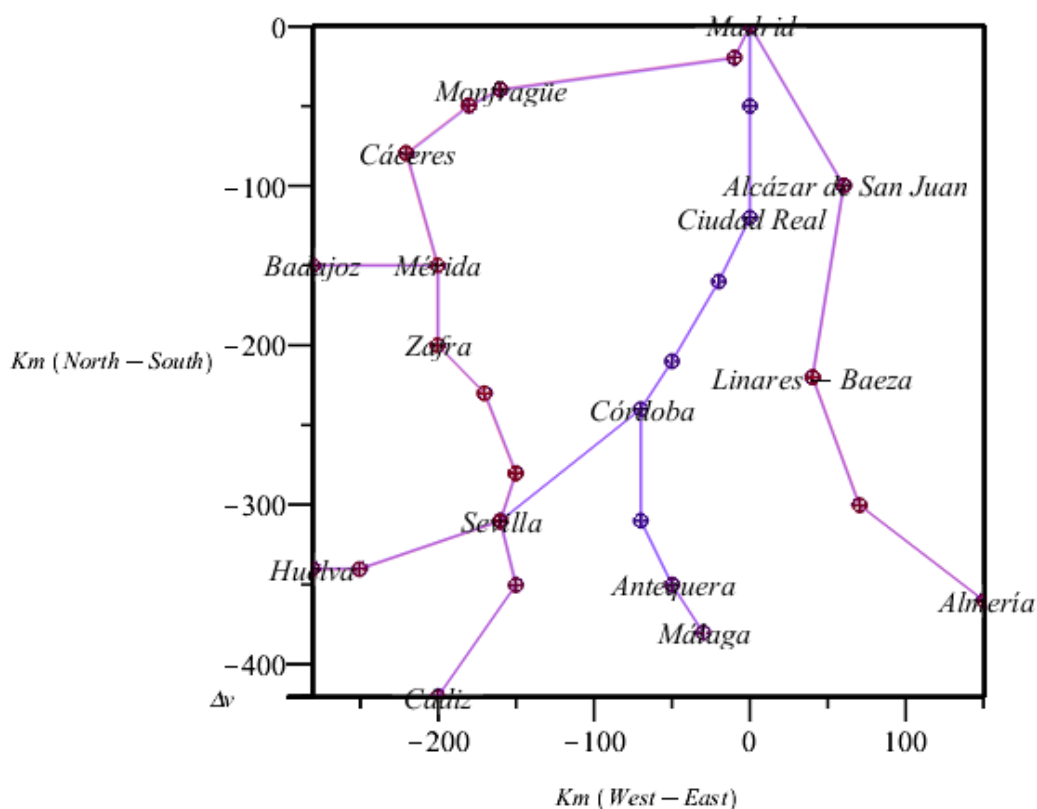
Although plotting rectangles whose areas represent time is fair, we are more used to directly represent speeds (or increases of speed) in the vertical axis. If this is what we want to represent, the data have to be transformed, what is performed by:

```
> DataDif():
```

and then, the lines of code below generate figures analogous to those above. For instance:

```
> `3Dtrains`:= GraphicR([op(AVv), op(IBv)]);
> PlotR(`3Dtrains`, [-90, 0, 0], 'pointline', 'B');
```

produces the zenithal view of the 3D plot of the example considered following Method II (Figure 9). Obviously, in the zenithal view there are no differences with respect to the 2D representation following Method I (Figure 3).



**Figure 9.** Zenithal view (Method II: increases of speed).

Observe in the two previous lines of code:

- the 'v' added to the names of variables 'AV' and 'IB'
- the final input to PlotR ('B' instead of 'A').

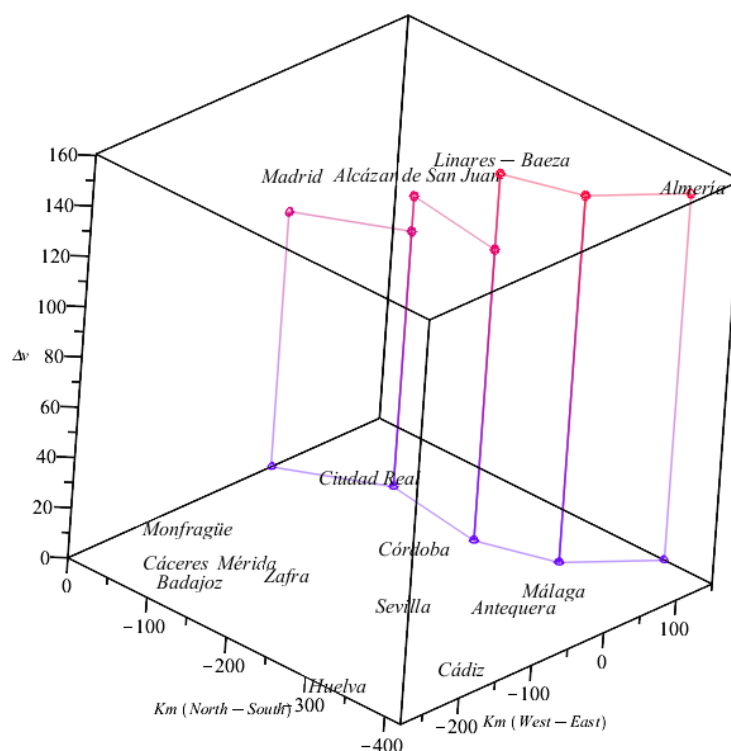
Meanwhile, the 3D plot of the timetable of the Madrid–Almería Talgo (alone) can be produced using Method II by typing:

```
> `3Dtrains`:= GraphicR([], IBv[2]):
> PlotR(`3Dtrains`, [-130, 62, 8], 'pointline', 'B');
```

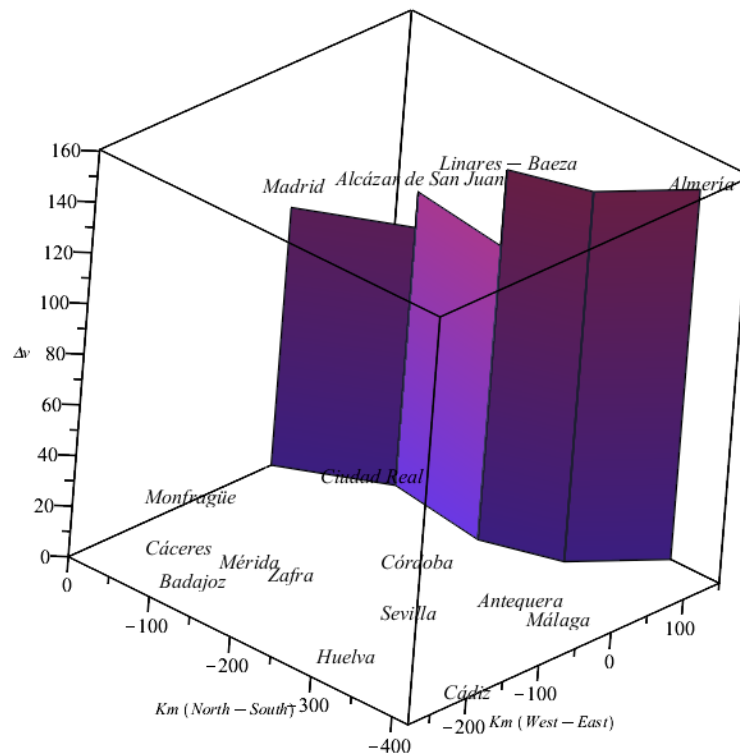
(see Figure 10) and the rectangles can be filled by typing

```
> PlotR(`3Dtrains`, [-130, 62, 8], 'surfacewireframe', 'B');
```

instead (Figure 11).



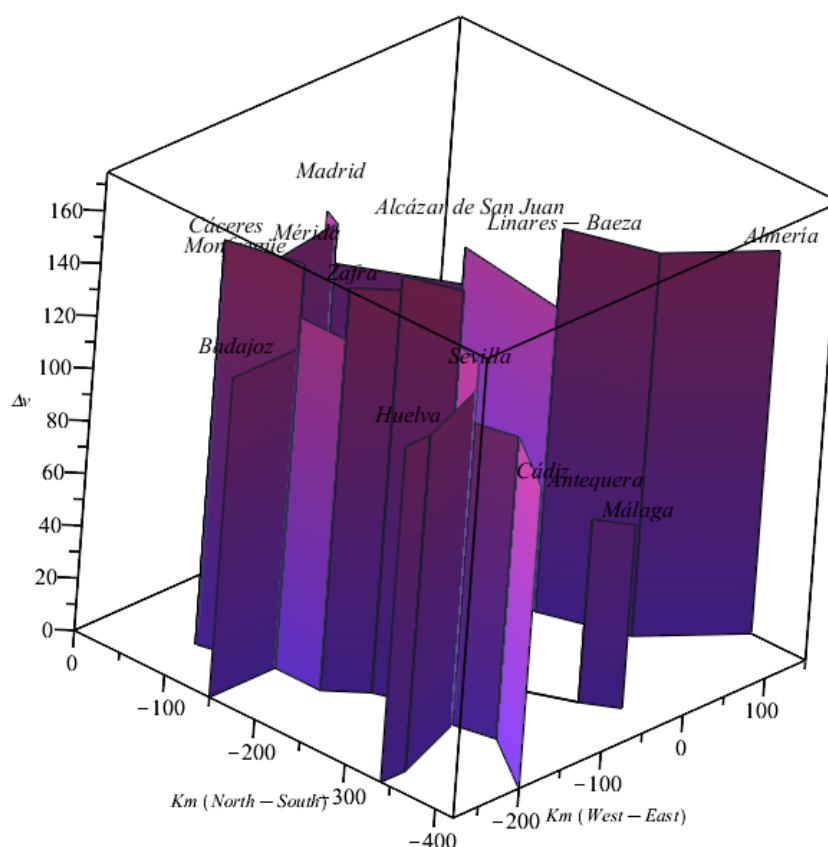
**Figure 10.** Frame 3D representation of the timetable of the Talgo train Madrid–Almería using Method II (note the difference in the “speed jumps” between adjacent rectangles with respect to Figure 4).



**Figure 11.** Filled rectangles 3D representation of the timetable of the Talgo train. Madrid–Almería using Method II (like Figure 10, with “filled” rectangles).

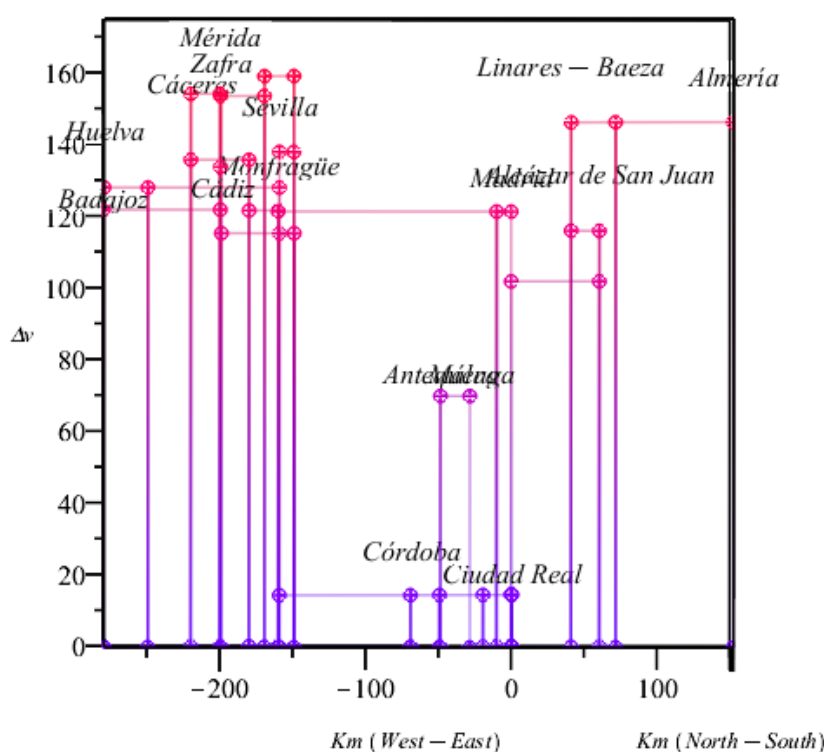
Finally, the 3D plot of the timetables of all the trains considered in the example (following Method II) from the same orientation (Figure 12) and from a side point of view (Figure 13) can be obtained by typing:

```
> `3Dtrains`:= GraphicR([op(AVv), op(IBv)]):
> PlotR(`3Dtrains`, [-130, 62, 8], 'surfacewireframe', 'B');
> PlotR(`3Dtrains`, [-90, 90, 0], 'pointline', 'B');
```



**Figure 12.** Filled rectangles representation of the speeds of all trains in the example along their corresponding routes (Method II).

The differences between both methods are most clearly visible in these two last plots. In Figure 12 it is clear that the maximum average speed of the network is reached in the central section between Córdoba and Málaga (the rectangle collapses into a segment on plane  $z=0$ ). The general differences are most clear in the side views (compare Figure 8 and Figure 13). As both methods have advantages, the choice is left to the user.



**Figure 13.** Side view of the frame 3D plot of all trains (Method II).

#### 4. Discussion

Data visualization can be very useful in many fields, including engineering in general and transportation engineering in particular. We believe this can be a curious tool for railway line and railway network speed data representation and comparison.

There are 3D approaches to visualization of graphs, but they usually deal with clearer representation of large graphs [31], clustering [32] and multilayer network visualization [33] (an approach also used in [34] by the authors). We know of no comparable software that uses the 3rd dimension to represent certain characteristics of a transportation network (the weight of the edges, from a graph theory point of view).

#### 5. Conclusion

The main computer algebra systems are extending their possibilities through the inclusion of packages for different purposes. They are becoming omnipresent in scientific computation. This article presents a novel application of graph theory to data visualization in railway engineering implemented on a computer algebra system. We believe that it can be very useful for presenting speed data in a visual way, something especially interesting for non-experts. Although the underlying mathematics is elementary, the implementation is complex.

## Acknowledgments

This work was partially supported by the Government of Spain under grant PGC2018-096509-B-I00.

We would also like to thank the anonymous reviewers for their detailed comments and suggestions.

## Conflict of interest

The authors declare no conflicts of interests.

## References

1. E. Roanes-Lozano, A. García-Álvarez, A. Hernando, *A geometric approach to the estimation of radial railway network*, Rev. R. Acad. Cienc. Exactas Fís. Nat., Ser. A Math., **106** (2012), 35–46.
2. E. Roanes-Lozano, J. L. Galán-García, A. García-Álvarez, et al., *Estimating radial railway network improvement with a CAS*, J. Comput. Appl. Math., **270** (2014), 294–307.
3. H. Small, *Florence Nightingale's statistical diagrams*, Stats & Lamps Research Conference organised by the Florence Nightingale Museum. St. Thomas's Hospital, 18th March, 1998.
4. M. van Campenhout, *Travel Time Maps*, Master's Thesis, Eindhoven: Technische Universiteit Eindhoven, 2010.
5. J. C. Denain, P. Langlois, *Cartographie en anamorphose*, Mappemonde, **49** (1998), 16–19.
6. N. Street, *TimeContours: Using isochrone visualisation to describe transport network travel cost*, Final Report, London: Department of Computing, Imperial College London, 2006.
7. M. Friendly, *The Golden Age of Statistical Graphics*, Stat. Sci., (2008), 502–535.
8. F. Galton, *On the Construction of Isochronic Passage-Charts*, Proceedings of the Royal Geographic Society and Monthly Record of Geography, **3** (1881), 657–658.
9. *Will California lead the way on trains?* The Economist, 2009. Available from: <https://www.economist.com/gulliver/2009/06/20/will-california-lead-the-way-on-trains>.
10. J. Gertner, *Getting Up to Speed*, New York Times, 2009. Available from: [http://www.nytimes.com/2009/06/14/magazine/14Train-t.html?pagewanted=all&\\_r=0](http://www.nytimes.com/2009/06/14/magazine/14Train-t.html?pagewanted=all&_r=0).
11. D. Dorling, *Anamorphosis: The geography of physicians, and mortality*, Int. J. Epidem., **36** (2007), 745–750.
12. Fundación de los Ferrocarriles Españoles. Available from: <http://www.ffe.es>.
13. H. Haverkort, *Embedding cues about travel time in schematic maps*, Schematic Mapping Workshop, 2014.
14. M. Cijssouw, M. M. Westenberg, *Embedding Travel Time Cues in Schematic Maps* (Master's Thesis), Eindhoven: Technische Universiteit Eindhoven, 2015.
15. E. Roanes-Lozano, E. Roanes-Macías, L. M. Laita, *An accelerated-time simulation of departing passengers' flow in airport terminals*, Mat. Comput. Simul., **67** (2004), 163–172.
16. A. Hernando, E. Roanes-Lozano, A. García-Álvarez, *An accelerated-time microscopic simulation of a dedicated freight double-track railway line*, Math. Comput. Mod., **51** (2010), 1160–1169.



17. G. Aguilera-Venegas, J. L. Galán-García, J. M. García, et al., *An accelerated-time simulation of car traffic on a motorway using a CAS*, Mat. Comput. Simul., **104** (2014), 21–30.
18. G. Aguilera-Venegas, J. L. Galán-García, E. Mérida-Casermeiro, et al., *An accelerated-time simulation of baggage traffic in an airport terminal*, Mat. Comput. Simul., **104** (2014), 58–66.
19. D. G. Zeitoun, T. Dana-Picard, *Accurate visualization of graphs of functions of two real variables*, Int. J. Appl. Math. Comput. Sci., **4** (2010), 960–970.
20. E. Bas, R. Ozarslan, *Theory of discrete fractional Sturm–Liouville equations and visual results*, AIMS Math., **4** (2019), 593–612.
21. A. Ciaramella, A. Staiano, *On the Role of Clustering and Visualization Techniques in Gene Microarray Data*, Algorithms, **12** (2019), 123.
22. G. Chartrand, L. Lesniak, P. Zhang, *Graphs & Digraphs*, Boca Raton–London–New York: CRC Press, 2010.
23. M. Abellanas, D. Lodaes, *Análisis de Algoritmos y Teoría de Grafos*, Madrid: RA-MA, 1991.
24. F. Dörfler, J. W. Simpson-Porco, F. Bullo, *Electrical Networks and Algebraic Graph Theory: Models, Properties, and Applications*, Proceedings of the IEEE, **106** (2018), 977–1005.
25. Y. Chen, P. Sareh, J. Yan, et al. *An integrated geometric-graph-theoretic approach to representing origami structures and their corresponding truss frameworks*, J. Mech. Design, **141** (2019).
26. A. Kaveh, K. Koohestani, *Graph products for configuration processing of space structures*, Comput. Struct., **86** (2008), 1219–1231.
27. E. Roanes-Lozano, L. M. Laita, *An applicable topology-independent model for railway interlocking systems*, Math. Comput. Simulat., **45** (1998), 175–183.
28. A. Martínez-Zarzuelo, E. Roanes-Lozano, M. J. Fernández-Díaz, *A Computer Approach to Mathematics Curriculum Developments Debugging*, EURASIA J. Math. Sci. Tech. Ed., **12** (2016), 2961–2974.
29. FuzzyMAD 2019. Available from: <https://eventos.ucm.es/43474/detail/fuzzymad-2019.html>.
30. G. Laporte, J. A. Mesa, F. A. Ortega, et al., *Locating a metro line in a historical city centre: application to Sevilla*, J. Oper. Res. Soc., **60** (2009), 1462–1466.
31. C. Ware, P. Mitchell, *Visualizing Graphs in Three Dimensions*, ACM T. Appl. Percept., **5** (2008), 1–15.
32. J. Lu, Y. Si, *Clustering-based force-directed algorithms for 3D graph visualization*, J. Supercomput., (2020), 1–62.
33. F. McGee, M. Ghoniem, G. Melançon, et al., *The State of the Art in Multilayer Network Visualization*, Comput. Graph Forum, **38** (2019), 125–149.
34. A. Almech, E. Roanes-Lozano, C. Solano-Macías, et al., *A New Approach to Shortest Route Finding in a Railway Network with Two Track Gauges and Gauge Changeovers*, Math. Probl. Eng., **2019** (2019), 1–16.

## Appendix

Code of the package (file rectangles.mpl)

```
# Variables:
Unitime    := 1/60:           #unit conversion (speed introduced km/min)
CotaI      := 1/(200*Unitime): #inverse of the max speed in the
                                #network (supposed > 200 km/h)
roll       := (2*rand(1 .. 5))/100: #random labels height (to avoid
                                #overlappings)

# Storing the data introduction
Introtrail := proc(lista, ancho)
  local n, p3, i, nodos, pulgares, WhoStation;
  global IB, AV, CotaI, stations;
  n := 1/2*numelems(lista) + 1/2;
  nodos := [];
  for i from 2 to n do
    nodos := [op(nodos), [op(lista[2*i - 3][1 .. 2]),
                          lista[2*i - 2]],
              [op(lista[2*i - 1][1 .. 2]), lista[2*i - 2]]];
  end do;
  if ancho = 'av' then
    AV := [op(AV), nodos];
  else
    IB := [op(IB), nodos];
  end if;
  p3 := x -> x[3];
  CotaI := min(CotaI, min(map(p3, nodos)));
  pulgares := SearchAll(1, map(x -> x[4], lista));
  WhoStation := map(x -> x[1 .. 3], lista[[pulgares]]);
  stations := stations union {op(WhoStation)};
end proc;

# Ordering data in order to apply polygonplot3d
polygons := proc(trail)
  local listado, i, up, down, n;
  listado := [];
  up := x -> [[x[1], x[2], 0], x];
  down := x -> [x, [x[1], x[2], 0]];
  n := numelems(trail);
  if 0 < n then
    for i to n - 1 do
      listado := [op(listado), [op(up(trail[i])),
                                op(down(trail[i + 1]))]];
    end do;
  end if;
end proc;
```

---

```

        end do;
    else
        listado := {};
    end if;
end proc;

# Labels matrix
InMatrix := proc(M, trail, origin)
    local x;
    global Unitime;
    for x in trail do
        M[x[2] + origin[2] + 1, x[1] + origin[1] + 1] := max(M[x[2] +
            origin[2] + 1, x[1] + origin[1] + 1], Unitime*x[3]);
    end do;
end proc;

# Scale of the gaps (e.g. 10 km)
Scale := x -> [partition*x[1], partition*x[2], Unitime*x[3]]:

# Setting a set with two plots elements (polygons & labels)
GraphicR := proc(trails)
    local i, trailEsc, noceroM, L, Listapolygons, stationsh, D0, D1;
    global stations, M, roll, graficos, ordinate, abscissa;
    M := Matrix(ordinate + 1, abscissa + 1, fill = 0);
    for i to numelems(trails) do
        InMatrix(M, trails[i], origin);
        trailEsc[i] := map(Scale, trails[i]);
    end do;
    stationsh := map(x -> [partition*x[1], partition*x[2], M[x[2] +
        origin[2] + 1, x[1] + origin[1] + 1] +
        max(M)*roll(), x[3]], stations);
    graficos := {textplot3d(stationsh)};
    Listapolygons := [];
    for i to numelems(trails) do
        Listapolygons := [op(Listapolygons),
            op(polygons(trailEsc[i]))];
    end do;
    graficos := graficos union {polygonplot3d(Listapolygons)};
end proc;

# Formatting the plot
PlotR := proc(DD, orientacion, estilo, tipo)
    local OptionsPlot, EjeZ, colores;
    if tipo = 'A' then

```

---

```

    EjeZ := h/Km;
    colores := z;
else
    EjeZ := `&Delta;v`;
    colores := z;
end if;
OptionsPlot := shading = colores, orientation = orientacion,
               view = 0 .. 1.1*max(M), style = estilo, symbol =
               sphere, thickness = 1; display(DD, labels = [Km*'West
               - East', Km*'North - South', EjeZ], OptionsPlot);
end proc;

# Data transformation (Method I -> Method II: "v^(-1)-> v[max]- v"

DataDif := proc()
    local i, velocidad;
    global AV, IB, AVv, IBv, Unitime, CotaI;
    Unitime := 1/Unitime;
    velocidad := x -> [x[1], x[2], 1/CotaI - 1/x[3]];
    AVv := AV;
    IBv := IB;
    for i to numelems(AV) do
        AVv[i] := map(velocidad, AV[i]);
    end do;
    for i to numelems(IB) do
        IBv[i] := map(velocidad, IB[i]);
    end do;
end proc;

```



**AIMS Press**

© 2020 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)