World Journal of Information Systems 2(4) 1-14
DOI: https://doi.org/10.17013/wjis.v2i4.46

*Article*

# Shortest paths on the Web as computing medium, in randomly generated maps

**Miguel Casquilho [1,*], Eliana Costa e Silva [2], Cristina Lopes [3], André Espírito Santo [4] and João Luís de Miranda [5]**

[1] Department of Chemical Engineering, IST, Universidade de Lisboa (University of Lisbon), and CERENA, Centro de Recursos Naturais e Ambiente ("Centre for Natural Resources and the Environment"), 1049-001 Lisboa, Portugal

[2] CIICESI, ESTG, Polytechnic of Porto, Margaride, 4610-156 Felgueiras, Portugal

[3] CEOS.PP, ISCAP, Polytechnic of Porto, 4465-004 S. Mamede de Infesta, Portugal

[4] Department of Computer Science and Engineering, IST, Universidade de Lisboa, 1049-001 Lisboa, Portugal

[5] Department of Technologies, ESTG, I. Politécnico de Portalegre (Polytechnic Institute of Portalegre), and CERENA, 7300-555 Portalegre, Portugal

***** Correspondence: mcasquilho@tecnico.ulisboa.pt

(Article history: Received July 30, 2025; Received in revised form August 25, 2025; Accepted October 24, 2025; Available online December 15, 2025)

**Abstract:** When dealing with routes in maps, the shortest path is the commonest addressed problem, but the topology of the map affects the paths' behavior in many ways. A study was conducted on random maps, by Monte Carlo simulation, in which the weights are uniformly distributed, in order to assess the impact of these conditions on the shortest paths. We provide a freely accessible webpage, where the user can insert the number of nodes, specify whether it is a directed or undirected graph, and the range of the distribution, to observe the influence of each parameter. Authors consider several goals: Offer the procedure computation, based on the well-known Dijkstra's algorithm; Combine the PHP language with a numerical language (Fortran), with the gnuplot graphing utility; and stress the suggestion of the Internet as a computing medium. The simulation runs on the webpage, thus needing no software installation or special power or matching operating system. This Web application is an example for many other problems, and is implemented on a Linux platform. The study draws attention both to the use of the Web for scientific computing, and to the convenience of this use in scientific publications. Authors advocate web-based computing in general, as it can use the same programs as classical computing, the programs being always the technical difficulty. In our technological era, this still scarcely explored approach also promotes the transfer of knowledge from academia to industry.



© 2025 ITMA. All rights reserved.

**Keywords:** Web-based application; random map; shortest path; Monte Carlo simulation; scientific computing; Engineering education

## 1. Introduction

Web-based computing provides resources made easily accessible simply through a webpage,[1] available just using a browser, to solve a problem of a given type. The data are supplied by the user to get a result. In this text, we describe a web based solver to find shortest paths in random maps, subject to a few constraints. In order to make the problem easily manageable, the data are, essentially: the number of nodes in a graph, or "cities"; whether the graph is directed or undirected; and the range of values, lower, L, and upper, U, of the uniform distribution, which simulates the weights of the edges. No prohibited edges are considered, so as to preclude the case of impossible results. Notwithstanding, as the weights will not correspond to Euclidean distances, a high U can, of course, act as (almost) infinity. In these assumptions, a simple map as a directed graph is suggested in Figure 1 for 4 cities, each geographical location being arbitrary, in this context.

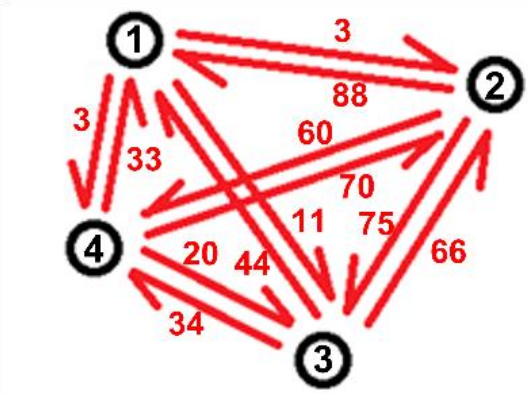


**Figure 1.** A random map with 4 cities, as a directed graph (edge weight values in Uniform[1, 100]).

The subject in this text belongs to 'Computer Science Engineering' or 'Informatics'[2], a term that we henceforth will use.

We consider a web-based solver as a program that needs no software installation by the user. This is the difference from the common applications, which (even if free) demand their installation, operating system compatibility, adequate power, and maintenance. The presence on the Internet simplifies the action, and invites collaborations, as insisted in the last two decades (e.g., [2]–[6]), namely often in previous editions of the CISTI (Conference).

[1] We are using "webpage", instead of "web page" (Web page, etc.), for lack of a reliable standard.

[2] We think it is time to use the terms "Informatics" and "informatic engineer" — from Russian "информатика" (informátika) ([1]). See also a further reference to Y. Ye [13].



© 2025 ITMA. All rights reserved.

There are several websites (e.g., [7], where data have to be inserted one by one, or [8], with unknown data format) and other algorithms to address the problem of the shortest path in a map.

Our contribution is to offer a free webpage for the informatic problem of finding the shortest path in random maps in an intuitive operation of a web-based program, for which we found no similar. A brief statistical analysis of the results is also given. The program will be explained regarding its architecture and use.

In our technological era, many people, wrongly, use the term "technological" instead of "informatic" or related to "computer science" (or information technology). Innumerable scientific articles, showing their numeric results, give the reader no confirmation thereof. Recently, publications are offering their programs, but possibly not matching our operating system or known languages (if free). In the 'FAIR' initiative, Wilkinson et al. [9] write: "There is an urgent need to improve the infrastructure supporting the reuse of scholarly data." The issue is the same (or more stringent) with the programs in the scientific literature. The editor of a statistics journal [10] wrote to "strongly encourage data and code sharing associated with newly submitted papers", which is a narrow advice. Indeed, it is the combination of both data and code that solves problems. Another serious observation is the one by Park et al. [11], who notice the flood of scientific publications with little added value, produced under the "publish or perish culture", of which (our opinion) one of the consequences is the disappearance of clear computing.

In the next Sections: "The problem", the simulation problem is briefly described; "Web applications" are mentioned, in view of their scarcity; and "Results and discussion" show the architecture of our application, with its results. Some "Conclusions" are finally presented.

## 2. The problem

The random maps problem is succinctly described as follows. If $n$ cities are connected by edges with weights in Uniform [$L$, $U$], what is expected for the distance of the shortest paths, i.e., the total weight, and for their length, i.e., number of edges?

We consider a random map of $n$ cities to be a complete weighted graph with $n$ nodes, where the edge weights are uniformly distributed in the (integer) range [$L$, $U$]. For the distance of the shortest path, at first sight, the worst case, $n\,U$, might be little probable, while the best case, $n\,L$, would be rare. We apply the well-known Dijkstra's algorithm to many, $N$, samples of random graphs with $n$ cities, and observe the frequency of the shortest paths' distances and length from city 1 to city $n$.

Although there are several other algorithms for solving shortest paths, once the aim of the present study is not to elaborate on shortest path algorithms, we will use Dijkstra's, for its simplicity and applicability. Dijkstra's algorithm is interestingly narrated and commented by Gries [12], while Ye [13] describes some of its variants.



© 2025 ITMA. All rights reserved.

Our application produces a square matrix with random values in [$L$, $U$], and finds the shortest path from city 1 to city $n$. Thus, the edges that constitute the path (typically given by the algorithm) are not taken into consideration, as the variables of interest are the shortest distance and the length of the path.

We adopted a code, from Burkardt's prolific repository [14], that deals with integer edge values. The complete result of the computing in a map typically includes the distances from city 1 to all the others, but, as mentioned, we take only the distance from city 1 to city $n$, the last in the list, as well as the number of edges. This is done a large number of times, by default $10^6$, to take some statistical conclusions.

In a webpage of ours [15], an application is available to solve the map problem. Consider, for example, the graph presented in Figure 2, from [16]. We write "1 2 5" to denote that the weight of the edge from city 1 to city 5 is 2. Therefore, the complete data for this graph is: 1 2 5, 1 3 2, 2 3 3, etc., as seen in Table 1. These data can be copied and pasted in the webpage, to obtain all the minimum distances and the best route.

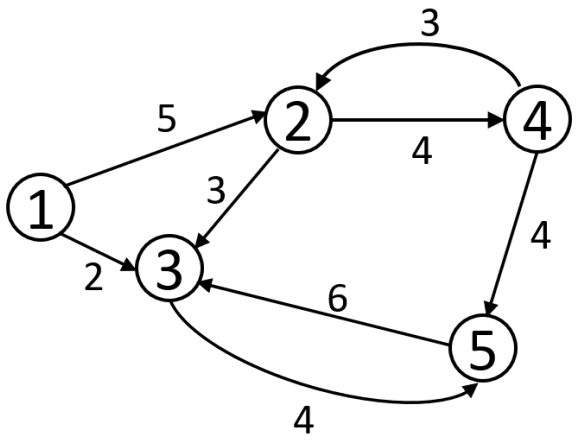


**Figure 2.** A random Example (from [16], p. 16).

**Table 1.** Data and results for the example of Figure 2.

| 1 2 5, 1 3 2, 2 3 3, 2 4 4, 3 5 4, 4 2 3, 4 5 4, 5 3 6 |
|---|
| Route from node 1 to node $n$: 1—3—5 |
| Shortest distance: 6 |

Regarding our present study, the webpage (given in the next Section) makes a Monte Carlo simulation to investigate the statistical behavior of: (*i*) the shortest distance: and (*ii*) the number of edges (length) in the simulated shortest distances.

In the Section "Results and discussion", we detail the computation, mainly in its programming on the Internet. This is readily accessible and able to be computed at the respective webpage, as explained further, other webpages of ours being available for other problems.



© 2025 ITMA. All rights reserved.

## 3. Web applications

The present study shows a web-based application that does a brief statistical analysis of a problem of random maps:

Ref. [17] [3]

This constitutes an inherent suggestion for the use by students in Science, Technology, Engineering, Mathematics (STEM).

As an example of the Web as a computing medium as we understand it, we use to mention Ponce ([18], in Hydraulics), though many websites, of course, permit computations by the user. That site provides, since 2004, a "virtual laboratory" ([19], which has a concept similar to ours, started in 1998. For an arbitrary example of his problems, see Figure 3, "Sequent depth in a hydraulic jump", Problem No. 111 ([20]). Vlab's user inserts the input data and gets the results. In our web pages, we always give default data (absent there).

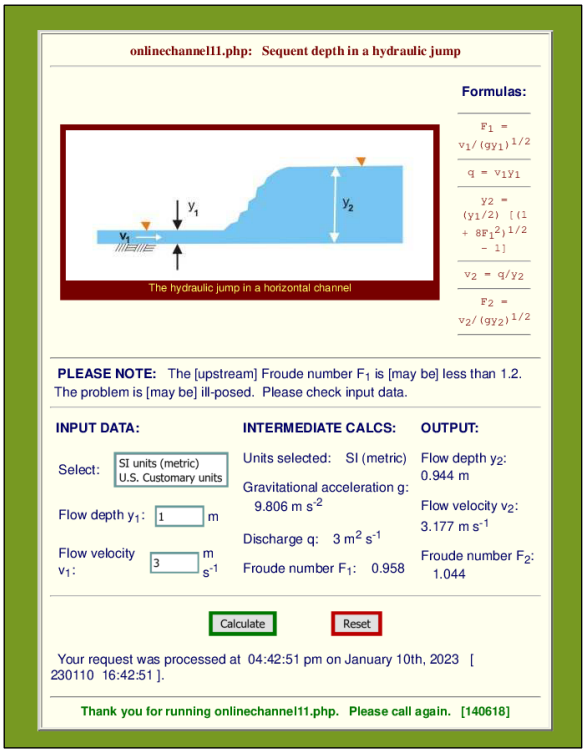


**Figure 3.** Ponce's Problem No. 111.

Our web pages (or Ponce's) do not discourage studying. From our more recent experience, Engineering students have quit programming. If the professors (ourselves) do not use real computing, the students fall to poorly used Excel. With webpages to solve the problems, students should do their own calculations, comparing results with those from a webpage, such as we are presenting ([17] again). Many other cases of ours appeared in previous editions of the CISTI Conference, as already mentioned, and in literature [21].

[3] I.e., https://web.tecnico.ulisboa.pt/~mcasquilho/compute/CISTI_2023/Randmap/Fx-randMap.php



© 2025 ITMA. All rights reserved.

The students avoid the "new technologies", but they are not alone. In the scientific literature, we rarely find a scientific article on numerical topics giving a webpage with their computation where we might use it.

The existence of Computer Science standards facilitates the combination of various languages and utilities, with their own characteristics. In this study, we combine Fortran 90 with 'gnuplot', under PHP, all of them free. The present work leads to intensive computing, as happens with Monte Carlo most of the time, so Fortran (compiled, fast) seems a reasonable choice. We already mentioned suggesting this example as a template, as we advocate the Web for computing.

For Fortran, we refer to the Intel compiler, version 11. For 'gnuplot', we use version 5.4. These versions (not our choice) are those installed by the administration in the public Linux system in our university. Python ([22], [23]) might have been used, even if using some Fortran subprograms for speed; indeed, calling Fortran from Python is usually very easy and straightforward.

## 4. Results and discussion

Among various tested combinations of the problem parameters: if the range [$L$, $U$] is wide enough, the distances tend to be small (near the absolute minimum, $n\,L$); and surprisingly the lengths of the shortest paths are also small.

As this work is for use on the Web, the computing itself is not visible to the user, and is executable at the said [17]. There, the user has available a set of default data, and just clicks the "Execute" button, or inserts a particular problem data (input) and gets the corresponding results (output). The results are built dynamically in HTML through the scripting language PHP, "native" for the Web [24]. The architecture has two components: a web-based front-end, to show the user interface, and a computing back-end to execute the tasks.

The front-end accepts the task's arguments (data), processes them, schedules the task, and sends these data to the back-end computing system. The back-end finishes the task and replies to the front-end with the output in a dynamic new page. (The results might appear in the same page, through PHP empty 'action', but we found that option to be less clear.) In the present case and in this type of problems, the output is simply alphanumeric with graphics, as frequently in Engineering. This structure is common to our problems, because our intention is to solve them, and make them available.

The numerical segments of our problem are solved in Fortran, which has no native graphical capabilities. This might lead to use Python (at least as manager), but we called 'gnuplot' [25], a free graphical package, to which we are more accustomed than Python plotting libraries, such as 'matplotlib.pyplot'. Additionally, we consider it a good decision to use 'gnuplot' as it turns out being a general tool, callable from many languages, including of course Python.



© 2025 ITMA. All rights reserved.

For the Web, the results of the problem are formatted with HTML (essentially with the tag ‘pre’, i.e., pre-formatted text, intermixed with images), and shown to the user.

The example is run on the public system of CIIST, the university’s Computing Centre [28], with a Linux operating system. As of the writing date, it reports 64 GB memory; ‘uname’ gives amd64, Debian 5.10.149 (Oct. 2022), x86_64 GNU/Linux; and ‘lscpu’, 8 CPU’s, from Xeon, 2 GHz.

The architecture follows these steps, where the names (bold, italics) follow the particular case described:

a) The user addresses the problem’s PHP web page, ***Fx-randMap.php***, and inserts the data (input) in the fields of an HTML ‘form’ in the page (data sent to the next file through PHP POST variables) (“F” recalling a Fortran executable as computing engine).
b) Upon clicking the ‘Execute’ button, the data are sent to an intermediate PHP file, from the ‘form’ statement ‘action=***RandMap.php***’.
c) This PHP file prepares the input received and sends it to the previously prepared executable (next) as command line arguments. The PHP file takes the results from the execution, and creates a (temporary, dynamic) PHP file (webpage) with the results.
d) The standalone executable, ***randMap.exe***, is then run through a PHP ‘shell_exec’ statement. This runs the calculations.
e) The output (to *stdout*) is sent to a temporary, dynamic page, the Results webpage. This page contains an HTML ‘pre’ tag with the text and one or more graphics (png’), which were just generated through a system call to Linux ‘base64’ utility. This way, no files are left on the server, needing no further deletion. This also avoids clashes between several users addressing the webpage, a fact to be considered in public access.
f) Auxiliary files — PHP environment files through PHP ‘include’, i.e., cascading style sheet, and the constant images that characterize the website.

The files necessary to implement the above actions are also mentioned for conciseness in the webpage.

In the above, the emphasized file names obviously vary depending on the particular problem, while, for the sake of simplicity and consistency, other files are the same for other problems. Schematically, Figure 4 shows the structure.



© 2025 ITMA. All rights reserved.

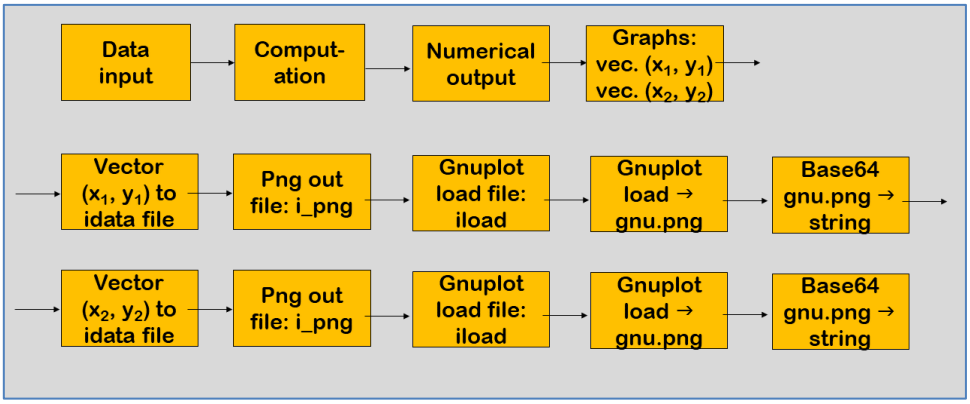


**Figure 4.** Communication among the various webpage components.

The fact that the problem runs "over the Web" adds no difficulty versus the ordinary resolution of a scientific problem. Indeed, we typically run this type of application out of the Internet, previously. In a technological problem, as always, the difficulty is the inherent knowledge of the subject and the methods to solve it.

The language used in this application is Fortran 90, and the 'gnuplot' utility for plotting. The absence of licensing for these choices is also important, considering the presence on the Internet.

The data for the webpage are, together with their default values (in bold):

- Number of nodes ("cities"), $n$ = **20**;
- Type of graph, undirected or **directed**;
- Lower and upper limits for the edges' weights, $L$ = **1**, $U$ = **100**.
- Number of Monte Carlo runs ("trials"), $N$ = **1.e+5**.
- Random number generator seed, **75357**.
- Histograms class width, **1**.
- "Sample", **0** (no sample), or a number in [1, $N$] if a sample graph is desired (for independent verification).
- Boolean showing plots' data.

As default data for the problem are given to the user in the web page, results appear with a simple click on the 'Execute' button. An image of the input webpage is in Figure 5.



© 2025 ITMA. All rights reserved.

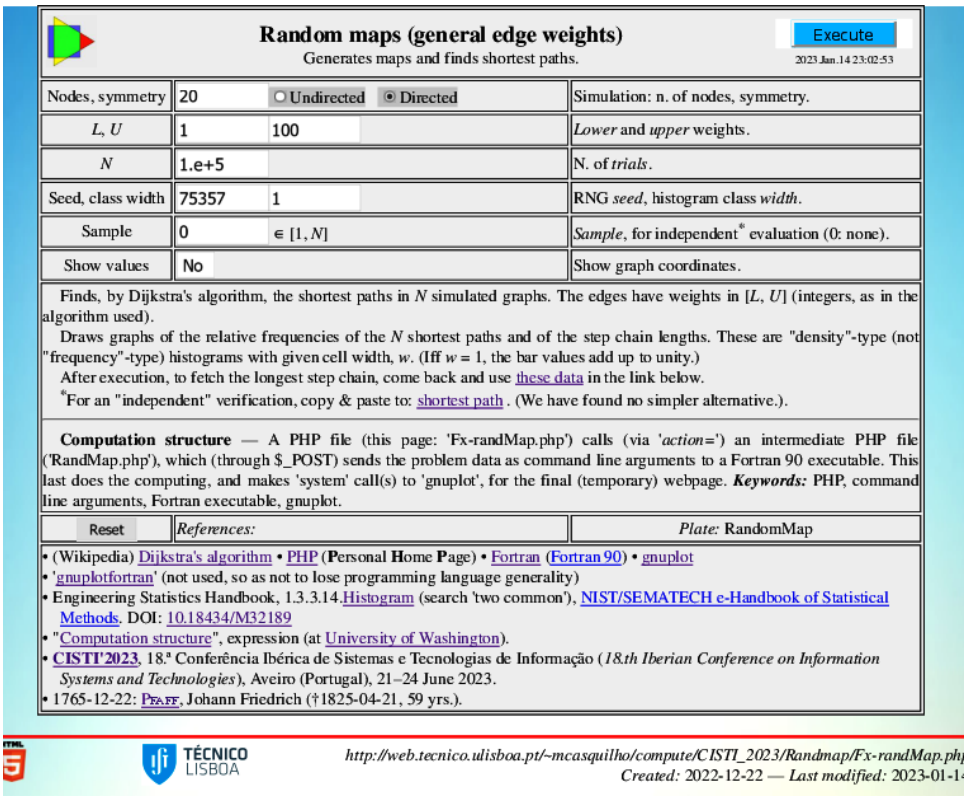


**Figure 5.** Web page (input) for the random maps problem. (The default option "Directed" is selected.)

The results of the execution are reproduced for the default data in

Figure 7, i.e., $(L, U) = [1, 100]$ and $10^5$ trials. In the horizontal axis of the upper graph are shown the values of the shortest distances of all the trials, concentrated around 20 with a Gaussian-like profile. This was executed in 1.1 s. The lower graph presents the behavior of the length (number of edges) of the shortest paths. This number is surprisingly low (also in Figure 7 b), certainly due to considering complete graphs.

In Figure 8 *a)* and *b)* are (partially) shown the results of a run with $(L, U) = [30, 100]$, showing a qualitatively different profile for both the shortest distances and lengths in the simulated shortest paths. The difference in parameter $L$ is certainly responsible for this behavior.

These significant differences show that the problem addressed can, out of the scope of the present text, lead to useful assessment if explored in its various parameters and with other probability distributions, such as a truncated Gaussian.



© 2025 ITMA. All rights reserved.

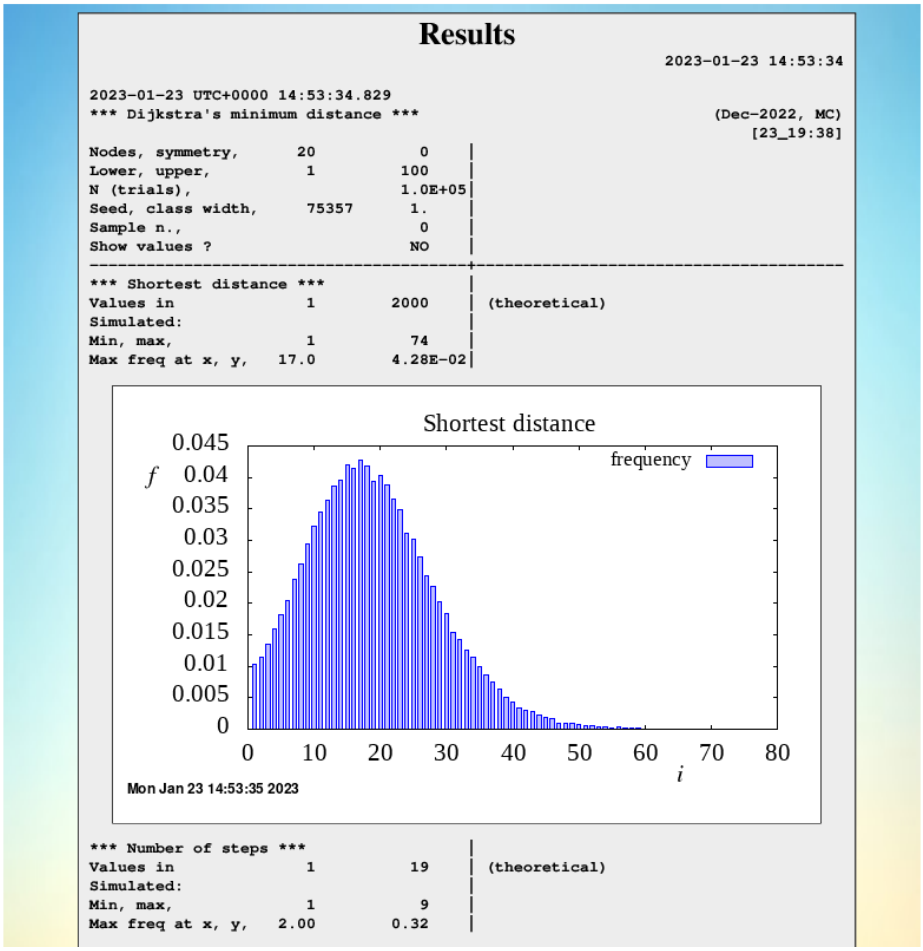


**Figure 6. (a)** Behavior of the shortest distances

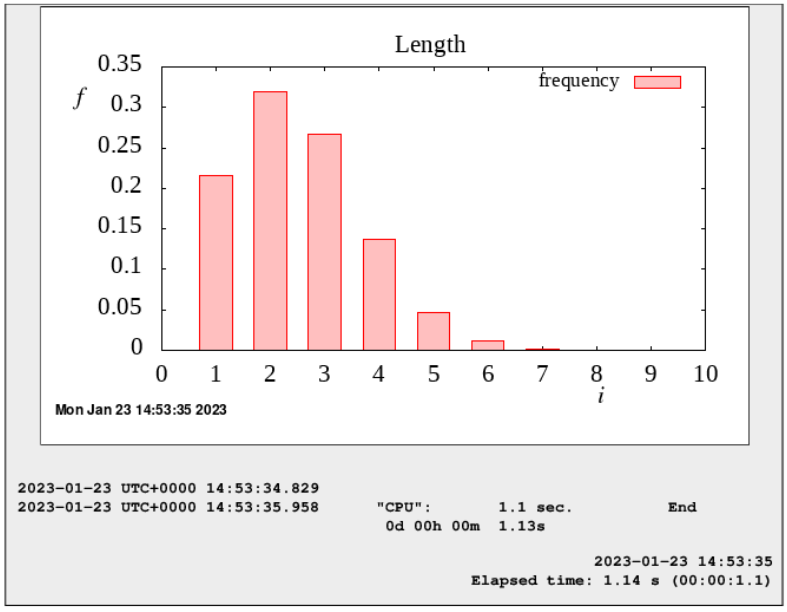


**Figure 7. (b)** lengths for the default problem.



© 2025 ITMA. All rights reserved.

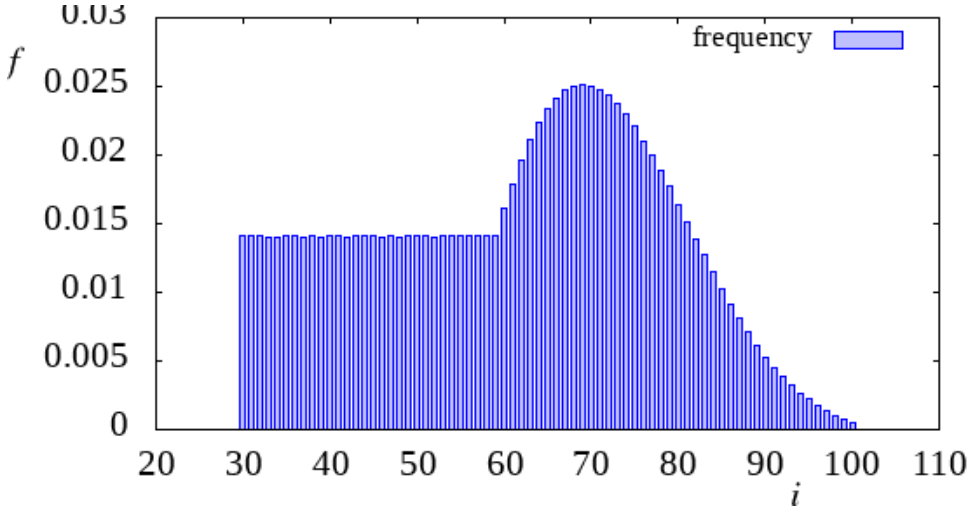


**Figure 8.** ***(a)*** Behavior (image) of the shortest distances for ($L$, $U$) = [30, 100] and 1.e+7 trials (CPU time 1'30").

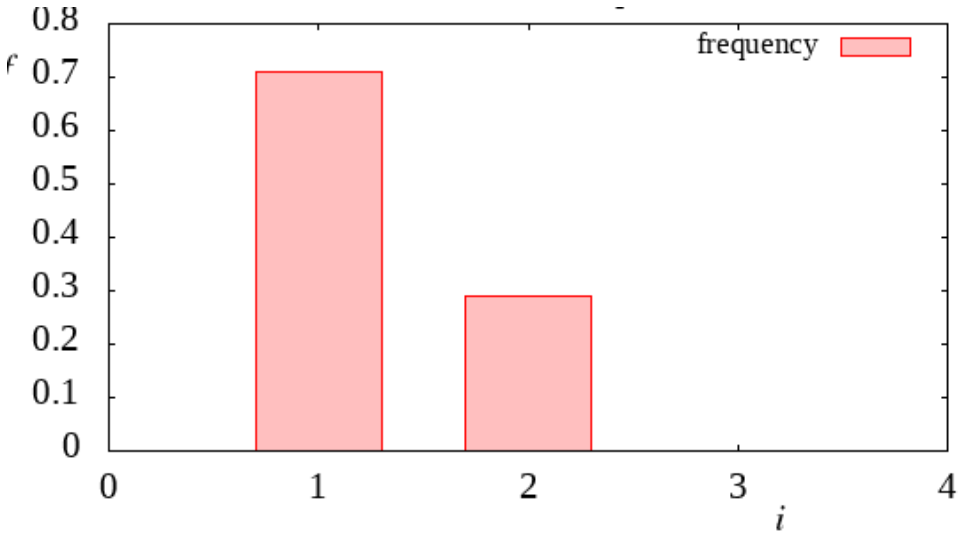


**Figure 8.** ***(b)*** Behavior (image) of the lengths for ($L$, $U$) = [30, 100] and 1.e+7 trials (CPU time 1'30").

## 5. Conclusion

In our opinion, there is still a high number of possibilities to use the Internet medium to explore scientific computing and in education. One comparable illustration of such computing by another author is mentioned, while an opposite panorama is shown by countless numerical publications, which offer no web-based verification. To this end, our applications only need a browser, dispensing with any software installation by the user, who also easily supplies the problem to solve. The illustrative application deals with a problem in constrained random maps, is programmed in PHP, Fortran 90 with 'gnuplot', and is available for execution on the authors' website. Our objectives were: to signal the use of this type of computing based on the client-server model, which permits access without any need from the user's terminal; to understand the shortest path problem, a classic problem in



© 2025 ITMA. All rights reserved.

Operational Research; assess the impact of its different parameters; and explore various statistical distributions; to draw attention of students (and authors) to computing; and to show advantages and ease of combining several languages, on the Internet environment.

The problem discussed tries to show the usefulness of the Web as a medium for scientific computing. Its exploration of the various parameters, here just skimmed over, appears surprisingly informative. The architecture proposed serves a great number of problems, for which the only difficulty is their intrinsic mathematical nature, while the language used can be any other, if compatible through the due present-day standards, and speed, security and licensing are adequate.

The programs and scripts used are essentially the same as the ones that would be used if run in a local computer, lending themselves to previous building and debugging. As we have verified in our academic and industrial practice, Web computing is recommendable for both purposes, favoring the connection and transfer between academia and industry.

**Acknowledgments:** The author MC does research at the Department of Chemical Engineering, IST, University of Lisbon, Lisbon, Portugal, and CERENA, "Centro de Recursos Naturais e Ambiente" (Centre for Natural Resources and the Environment), under Project UID/04028/2020, funded by FCT, "Fundação para a Ciência e a Tecnologia" (Portuguese National Science Foundation); ECS teaches and does research at CIICESI, ESTG, Polytechnic of Porto, supported by national funds through FCT, Project UIDB/04728/2020; CL teaches and does research at CEOS.PP, ISCAP, Polytechnic of Porto, financed by Portuguese national funds through FCT, under the Project UIDB/05422/2020; AES is MSc student at the Dept. of Computer Science and Engineering, IST; and JLM teaches and does research at the Dept. of Technologies, ESTG, Polytechnic Institute of Portalegre, Portalegre, Portugal, and CERENA. CIIST, "Centro de Informática do IST" (Informatics Centre of IST) supplies the computing system.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

[1] Lexico, 'informatic', https://www.lexico.com/definition/informatic , accessed 05-Jan-2023.

[2] B. Franco, and M. Casquilho (2011), "A Web application for scientific computing: combining several tools and languages to solve a statistical problem", CISTI'2011, *6.ª Conf. Ibérica de Sistemas e Tecnologias de Informação* (6.th Iberian Conference on Information Systems and Technologies), Chaves (Vila Real, Portugal.

[3] M. Casquilho (2013), *"Computação científica, Internet, Indústria"* (Scientific computing, Internet, Industry), 1.st Portuguese Meeting on Mathematics for Industry, FCUP, University of Porto, Porto (Portugal).

[4] M. Barros, and M. Casquilho (2019). Linear Programming with CPLEX: an illustrative application over the Internet", CISTI'2019, Coimbra (Portugal).



© 2025 ITMA. All rights reserved.

[5] M. Casquilho, J. Buescu (2022), “From unequal size sample sums to estimating the standard deviation”, ICMASC 2022, International Conference on Mathematical Analysis and Applications in Science and Engineering, Porto (Portugal).

[6] M. Casquilho, J. Buescu (2022), "Mean of Exponential distribution: estimation from sums of unequal size samples", COMPSTAT 2022, 24.th International Conference on Computational Statistics, Bologna (Italy), 23–26 Aug..

[7] Dijkstra's Algorithm Calculator, http://www.jakebakermaths.org.uk/maths/dijkstrasalgorithmsolverv9.html , accessed 05-Jan-2023.

[8] Technical University of Munich, Der Dijkstra-Algorithmus, https://algorithms.discrete.ma.tum.de/graph-algorithms/spp-dijkstra/index_en.html , accessed 05-Jan-2023.

[9] M. D. Wilkinson (2016), “Comment: The FAIR Guiding Principles for scientific data management and stewardship”, *Scientific Data,* 3:160018, 9 pp.

[10] B. M. Colosimo, ed. (2019), Message, *J. of Quality Technology*, 51 (1), 1–2 or https://www.tandfonline.com/doi/full/10.1080/00224065.2019.1569896., accessed 05-Jan-2023.

[11] M. Park, E. Leahey, R. Funk (2022), “Papers and patents are becoming less disruptive over time”, *Nature*, 613, 5 Jan 2023, pp 138–144 (doi: https://www.nature.com/articles/s41586-022-05543-x).

[12] D. Gries et al. (2012), “Shortest-path algorithm”, JavaHyperText and Data Structures, Department of Computer Science, Cornell University, https://www.cs.cornell.edu/courses/JavaAndDS/shortestPath/shortestPath.html , accessed 05-Jan-2023.

[13] Y. Ye (2019), Shortest path algorithms, Luddy School of Informatics, Computing, and Engineering, Indiana University, https://cgi.luddy.indiana.edu/~yye/c343-2019/shortest.php , accessed 05-Jan-2023.

[14] J. Burkardt (2010), “dijkstra”, https://people.sc.fsu.edu/~jburkardt/f_src/dijkstra/dijkstra.html , accessed 05-Jan-2023.

[15] M. Casquilho (2017), “Shortest path”, http://web.tecnico.ulisboa.pt/mcasquilho/compute/or/Fx-shortestpath.php .

[16] K. R. Pattipati (2016), Lecture 6: Shortest Path Algorithms: (Part I), Cyberlab Research, University of Connecticut, https://cyberlab.engr.uconn.edu/wp-content/uploads/sites/2576/2018/09/Lecture_6.pdf , accessed 05-Jan-2023.

[17] M. Casquilho (2022), “Random maps”, http://web.tecnico.ulisboa.pt/~mcasquilho/compute/CISTI_2023/Randmap/Fx-randMap.php .

[18] V. M. Ponce, San Diego State University, http://ponce.sdsu.edu/ , accessed 05-Jan-2023.

[19] V. M. Ponce (2017), Vlab, http://ponce.sdsu.edu/visualab.html , accessed 05-Jan-2023.

[20] V. M. Ponce (2017), Problem 111, http://ponce.sdsu.edu/onlinefallvelocity.php , accessed 05-Jan-2023.

[21] M. Casquilho, J. Buescu (2022), “Standard deviation estimation from sums of unequal size samples”, *Monte Carlo Methods and Applications* (doi: 10.1515/mcma-2022-2118).

[22] The SciPy community, “Using Python as glue”, https://numpy.org/doc/stable/user/c-info.python-as-glue.html , accessed 05-Jan-2023.



© 2025 ITMA. All rights reserved.

[23] T. E. Oliphant (2007). Python for scientific computing. *Computing in Science & Engineering*, *9*(3), 10–20, in Kermode (2020).

[24] PHP, The PHP Group, https://www.php.net/ , accessed 05-Jan-2023.

[25] Gnuplot, http://www.gnuplot.info/ , accessed 05-Jan-2023.

[26] CIIST, “Centro de Informática do IST” (Informatics Centre of Instituto Superior Técnico), https://ciist.ist.utl.pt/eng.php , accessed 05-Jan-2023.



© 2025 ITMA. All rights reserved.