



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

Print/export

[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Languages

[فارسی](#)
[Русский](#)

 [Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Marzullo's algorithm

From Wikipedia, the free encyclopedia

Marzullo's algorithm, invented by [Keith Marzullo](#) for his Ph.D. dissertation in 1984, is an **agreement algorithm** used to select sources for estimating accurate time from a number of **noisy** time sources. A refined version of it, renamed the "**intersection algorithm**", forms part of the modern [Network Time Protocol](#). The Marzullo's algorithm is also used to compute the [relaxed intersection](#) of n boxes (or more generally n subsets of \mathbf{R}^n), as required by several [robust set estimation](#) methods.

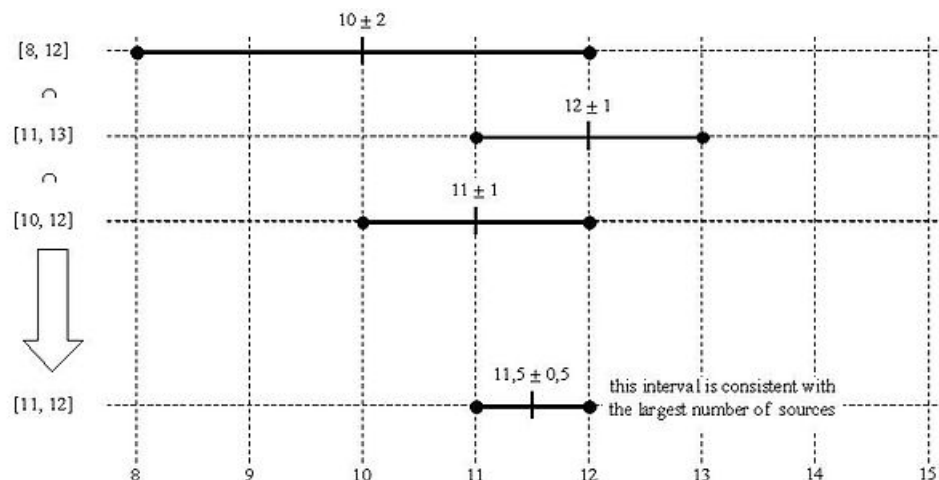
Contents [\[hide\]](#)

- [1 Purpose](#)
- [2 Method](#)
- [3 Efficiency](#)
- [4 References](#)
- [5 External links](#)

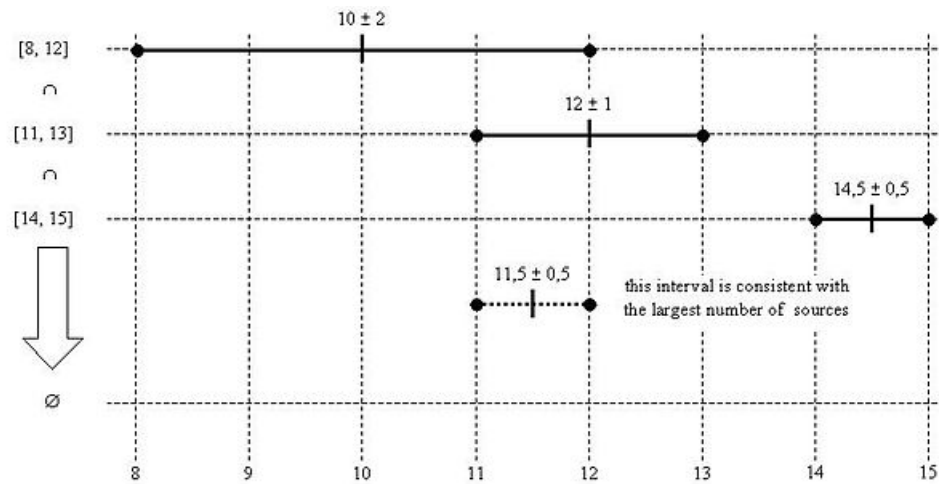
Purpose [\[edit\]](#)

Marzullo's algorithm is efficient in terms of time for producing an optimal value from a set of estimates with [confidence intervals](#) where the actual value may be outside the confidence interval for some sources. In this case the best estimate is taken to be the smallest interval [consistent](#) with the largest number of sources.

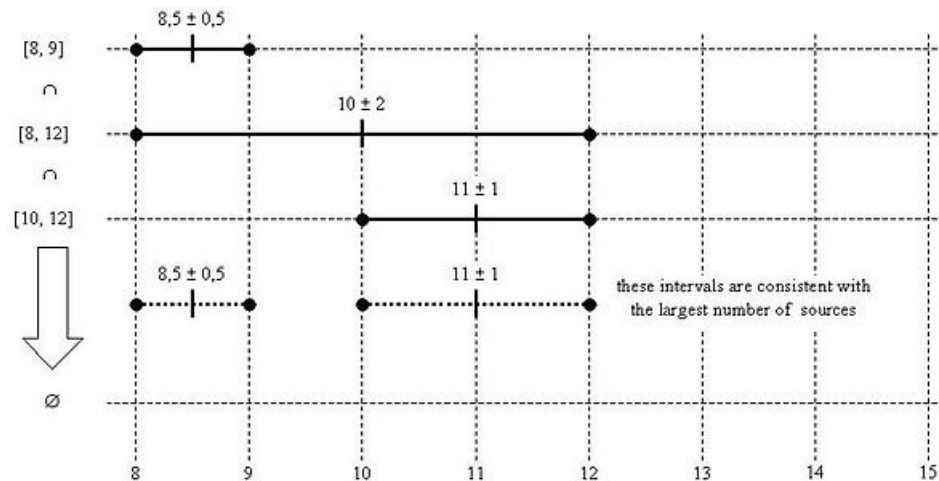
If we have the estimates 10 ± 2 , 12 ± 1 and 11 ± 1 then these intervals are $[8, 12]$, $[11, 13]$ and $[10, 12]$ which intersect to form $[11, 12]$ or 11.5 ± 0.5 as consistent with all three values.



If instead the ranges are $[8, 12]$, $[11, 13]$ and $[14, 15]$ then there is no interval consistent with all these values but $[11, 12]$ is consistent with the largest number of sources — namely, two of them.



Finally, if the ranges are $[8, 9]$, $[8, 12]$ and $[10, 12]$ then both the intervals $[8, 9]$ and $[10, 12]$ are consistent with the largest number of sources.



This procedure determines an interval. If the desired result is a best value from that interval then a naive approach would be to take the center of the interval as the value, which is what was specified in the original Marzullo algorithm. A more sophisticated approach would recognize that this could be throwing away useful information from the confidence intervals of the sources and that a [probabilistic model](#) of the sources could return a value other than the center.

Note that the computed value is probably better described as "optimistic" rather than "optimal". For example, consider three intervals $[10, 12]$, $[11, 13]$ and $[11.99, 13]$. The algorithm described below computes $[11.99, 12]$ or 11.995 ± 0.005 which is a very precise value. If we suspect that one of the estimates might be incorrect, then at least two of the estimates must be correct. Under this condition, the best estimate is $[11, 13]$ since this is the largest interval that always intersects at least two estimates. The algorithm described below is easily parameterized with the maximum number of incorrect estimates.

Method [\[edit\]](#)

Marzullo's algorithm begins by preparing a table of the sources, sorting it and then searching (efficiently) for the intersections of intervals. For each source there is a range $[c-r, c+r]$ defined by $c \pm r$. For each range the table will have two [tuples](#) of the form $\langle \text{offset}, \text{type} \rangle$. One tuple will represent the beginning of the range, marked with type -1 as $\langle c-r, -1 \rangle$ and the other will represent the end with type $+1$ as $\langle c+r, +1 \rangle$.

The description of the algorithm uses the following variables: best (largest number of overlapping intervals found), cnt (current number of overlapping intervals), beststart and bestend (the beginning and end of best interval found so far), i (an index), and the table of tuples.

1. Build the table of tuples.
2. [Sort](#) the table by the offset. (If two tuples with the same offset but opposite types exist, indicating that one interval ends just as another begins, then a method of deciding which comes first is necessary. Such an occurrence can be considered an overlap with no duration, which can be found by the algorithm by putting type -1 before type $+1$. If such pathological overlaps are considered objectionable they can be

avoided by putting type +1 before −1 in this case.)

3. [initialize] best=0 cnt=0
4. [loop] go through each tuple in the table in ascending order
 4. [current number of overlapping intervals] cnt=cnt−type[i]
 5. if cnt>best then best=cnt beststart=offset[i] bestend=offset[i+1]

commentary: the next tuple, at [i+1], will either be an end of an interval (type=+1) in which case it ends this best interval, or it will be a beginning of an interval (type=−1) and in the next step will replace best.

ambiguity: unspecified is what to do if best=cnt. This is a condition of a tie for greatest overlap. The decision can either be made to take the smaller of bestend−beststart or offset[i+1]−offset[i] or just take an arbitrary one of the two equally good entries.

6. [end loop] return [beststart,bestend] as optimal interval. The number of *false* sources (ones which do not overlap the optimal interval returned) is the number of sources minus the value of best.

Efficiency [edit]

Marzullo's algorithm is efficient in both space and time. The [asymptotic](#) space usage is $O(n)$, where n is the number of sources. In considering the asymptotic time requirement the algorithm can be considered to consist of building the table, sorting it and searching it. Sorting can be done in $O(n \log n)$ time, and this dominates the building and searching phases which can be performed in [linear](#) time. Therefore the time efficiency of Marzullo's algorithm is $O(n \log n)$.

Once the table has been built and sorted it is possible to update the interval for one source (when new information is received) in linear time. Therefore, updating data for one source and finding the best interval can be done in $O(n)$ time.

References [edit]

- K. A. Marzullo. Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service. Ph.D. dissertation, Stanford University, Department of Electrical Engineering, February 1984.

External links [edit]

- David L. Mills, [A Brief History of NTP Time: Confessions of an Internet Timekeeper](#) 
- [Keith Marzullo's homepage](#) 

Categories: [Agreement algorithms](#)

This page was last modified on 9 February 2014, at 00:26.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

