

Using AI to Find Worst-Case Execution Time

JUNE 15, 2023

Bjorn Andersson



This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

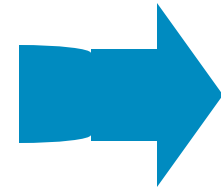
This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM23-0582

Using AI to Find Worst-Case Execution Time

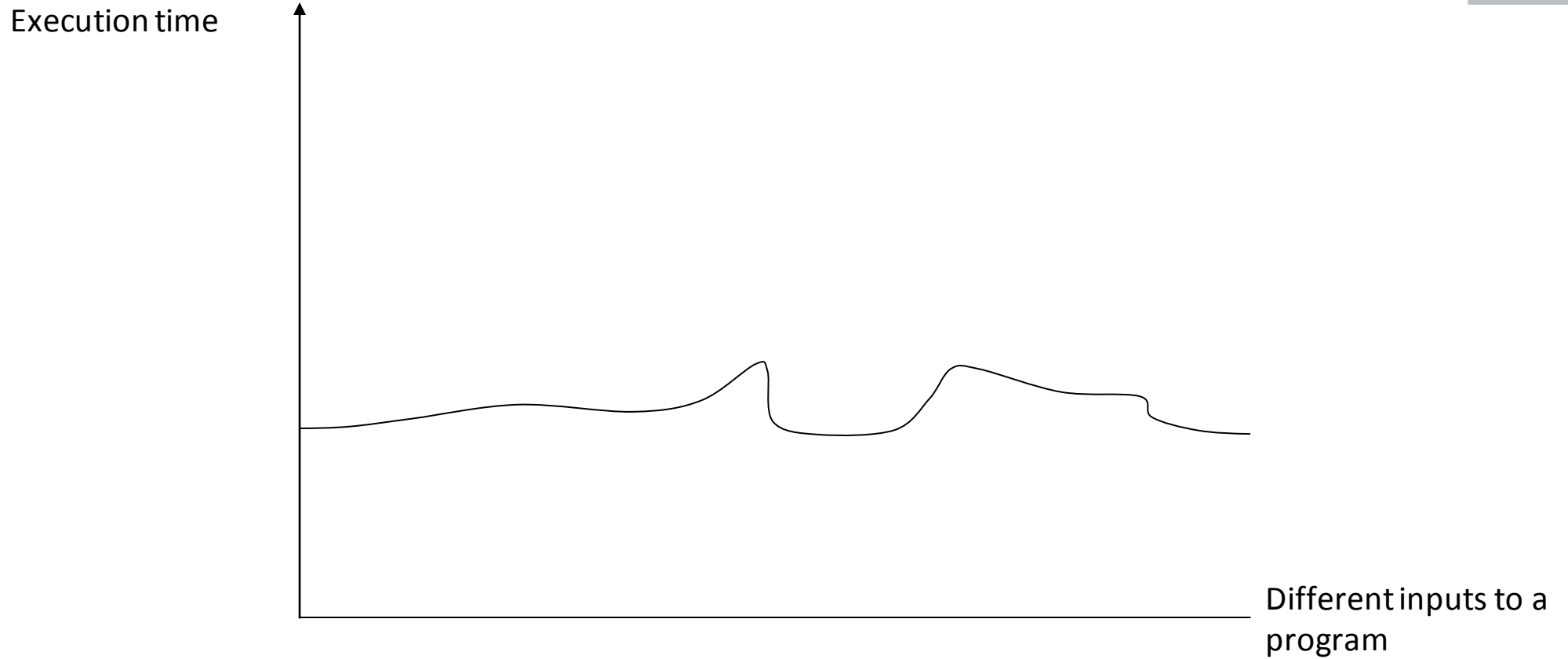
Convergence of Four Things

- WCET analysis is needed for certification of avionics software
- WCET analysis is becoming more challenging
- AI has shown great promise recently
- AI might help WCET analysis circumvent current limitations

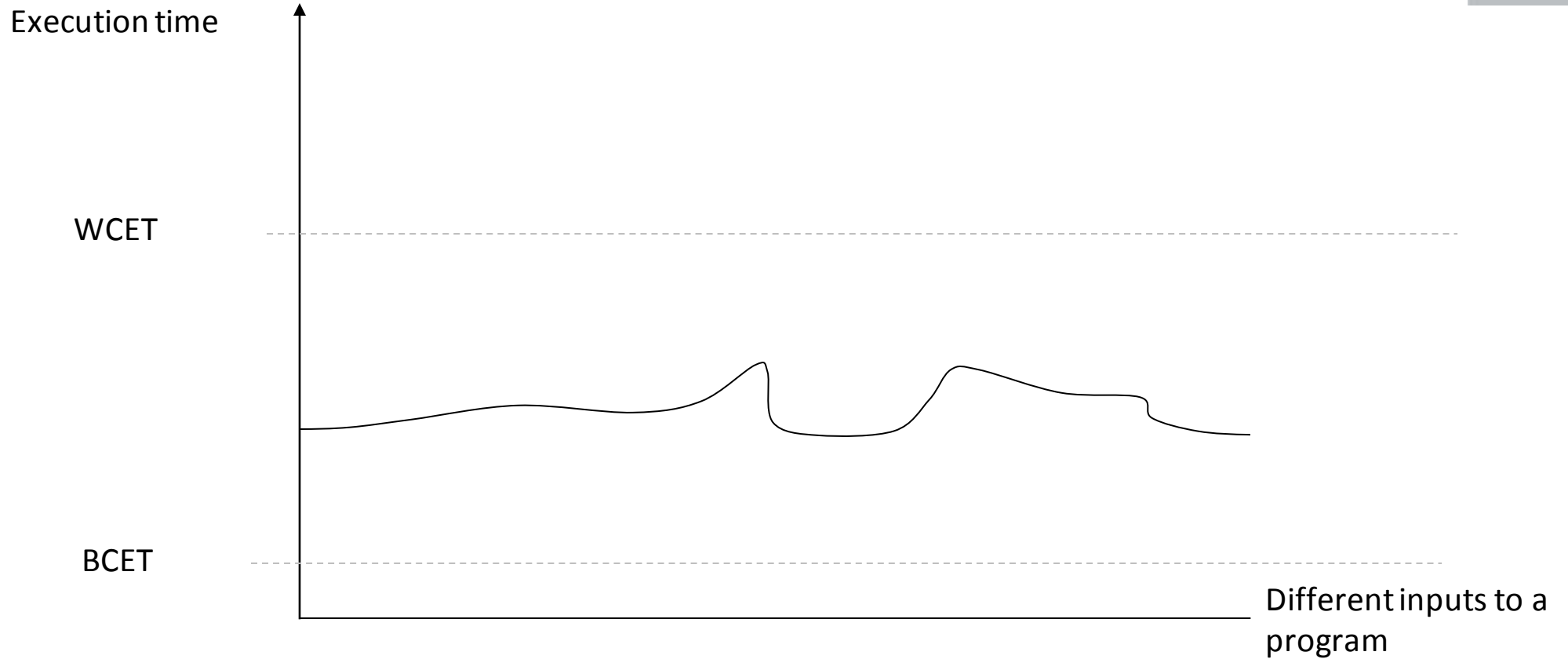


Can we use
AI for WCET
analysis?

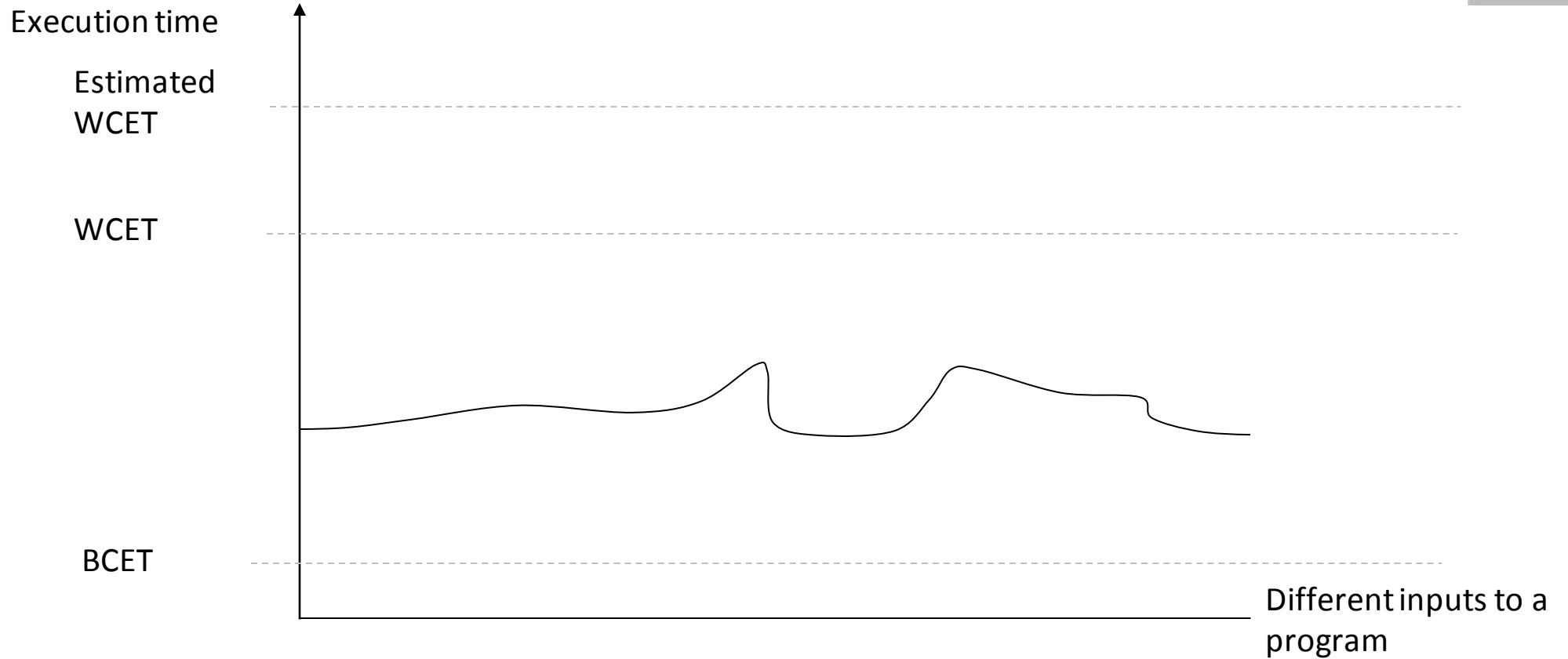
What is Worst-Case Execution Time?



What is Worst-Case Execution Time?

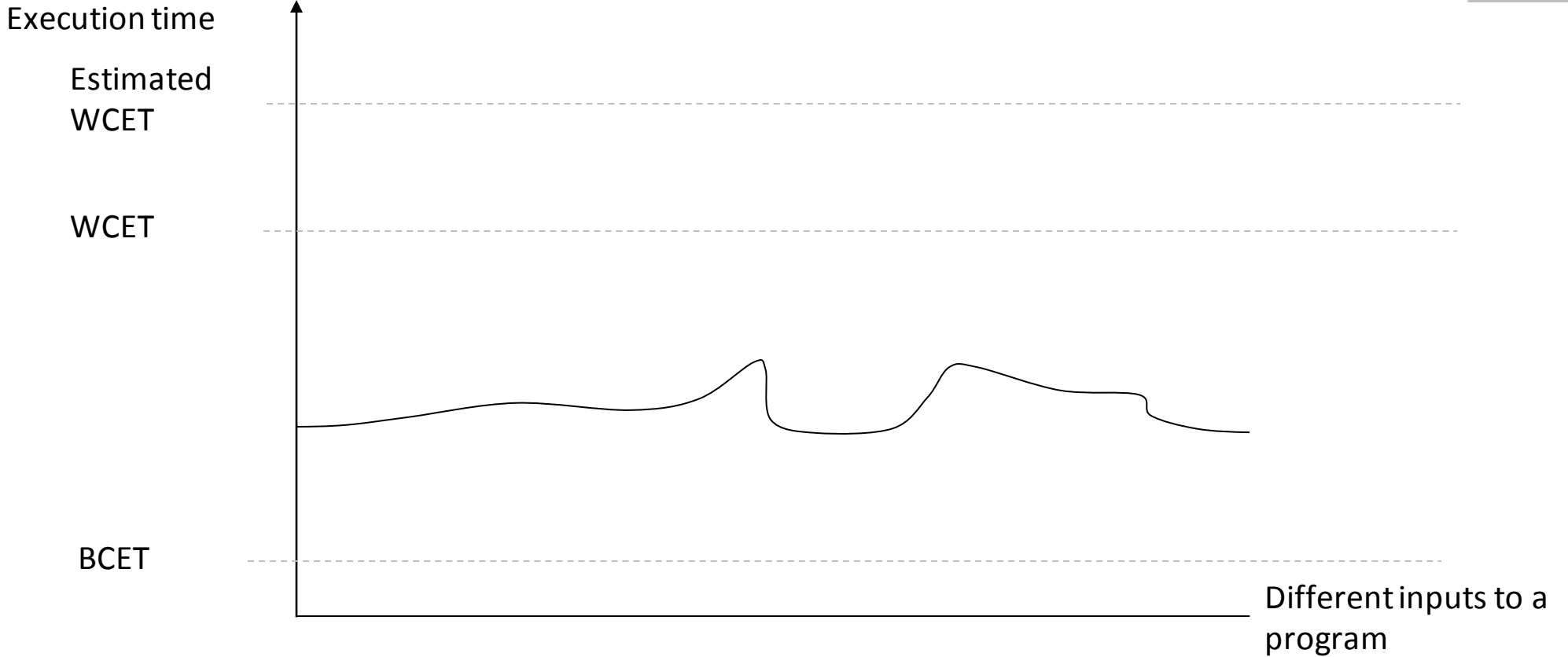


What is Worst-Case Execution Time Estimate?



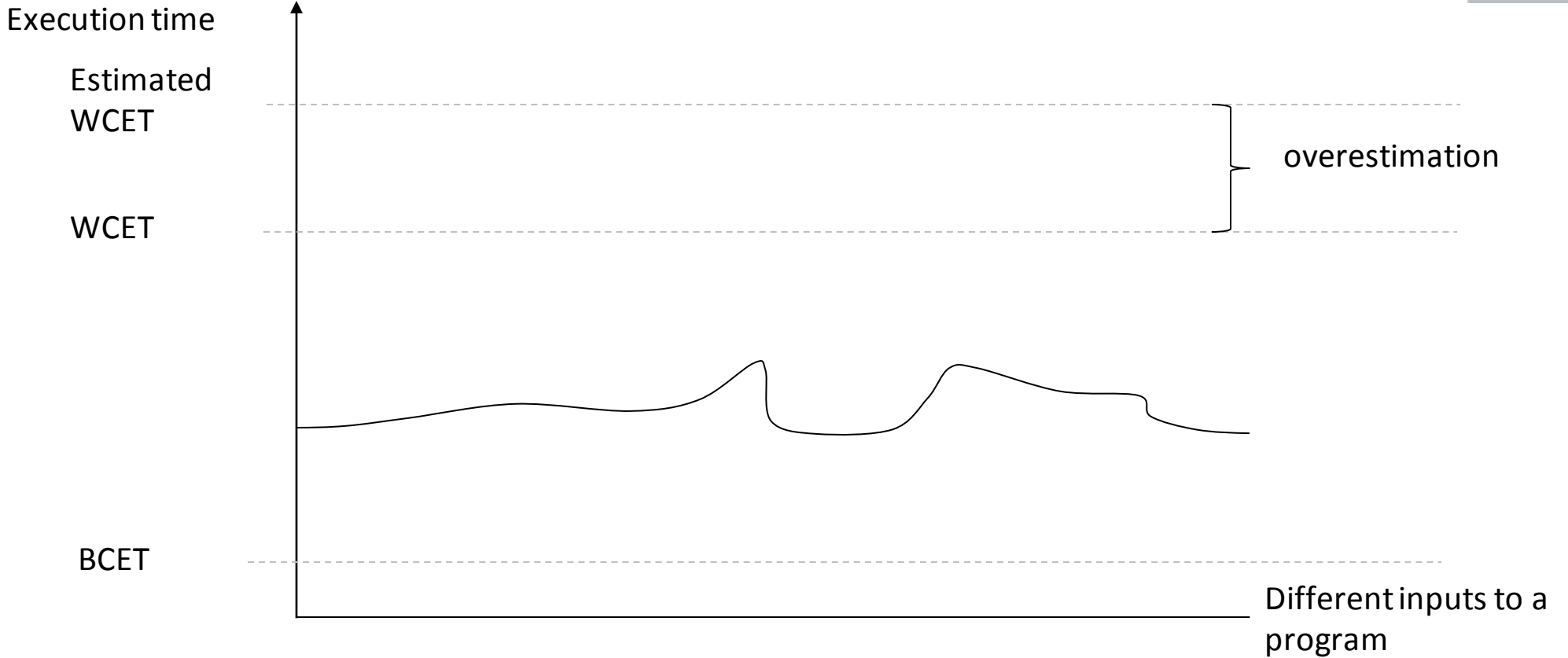
What is Worst-Case Execution Time Estimate?

Find WCET estimate



What is Worst-Case Execution Time Estimate?

Find WCET estimate with small overestimation



Why is Worst-Case Execution Time Analysis Challenging?

Software complexity

- The number of execution paths in a program tends to be very large and input dependent. We cannot explicitly enumerate all of them.

Why is Worst-Case Execution Time Analysis Challenging?

Software complexity

- The number of execution paths in a program tends to be very large and input dependent. We cannot explicitly enumerate all of them.

Hardware complexity

- Even for a single path in a program, the execution time depends on (i) initial state (variable initialization), (ii) state of the hardware (dirty cache blocks initially), (iii) behavior of hardware (cache, pipelining, etc).
- On a multicore, it gets even more complex because the execution time of a program depends on co-runners.

Current state-of-the-art/practice

Common practice

- Guess the worst-case input. Run the program with this input and measure execution time

Current state-of-the-art/practice

Common practice

- Guess the worst-case input. Run the program with this input and measure execution time

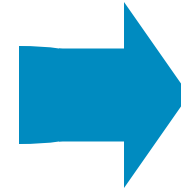
Academic research

- Static analysis: compute WCET without running the program
- Hybrid: measure some parts of a program and combine it with analysis

Current obstacles

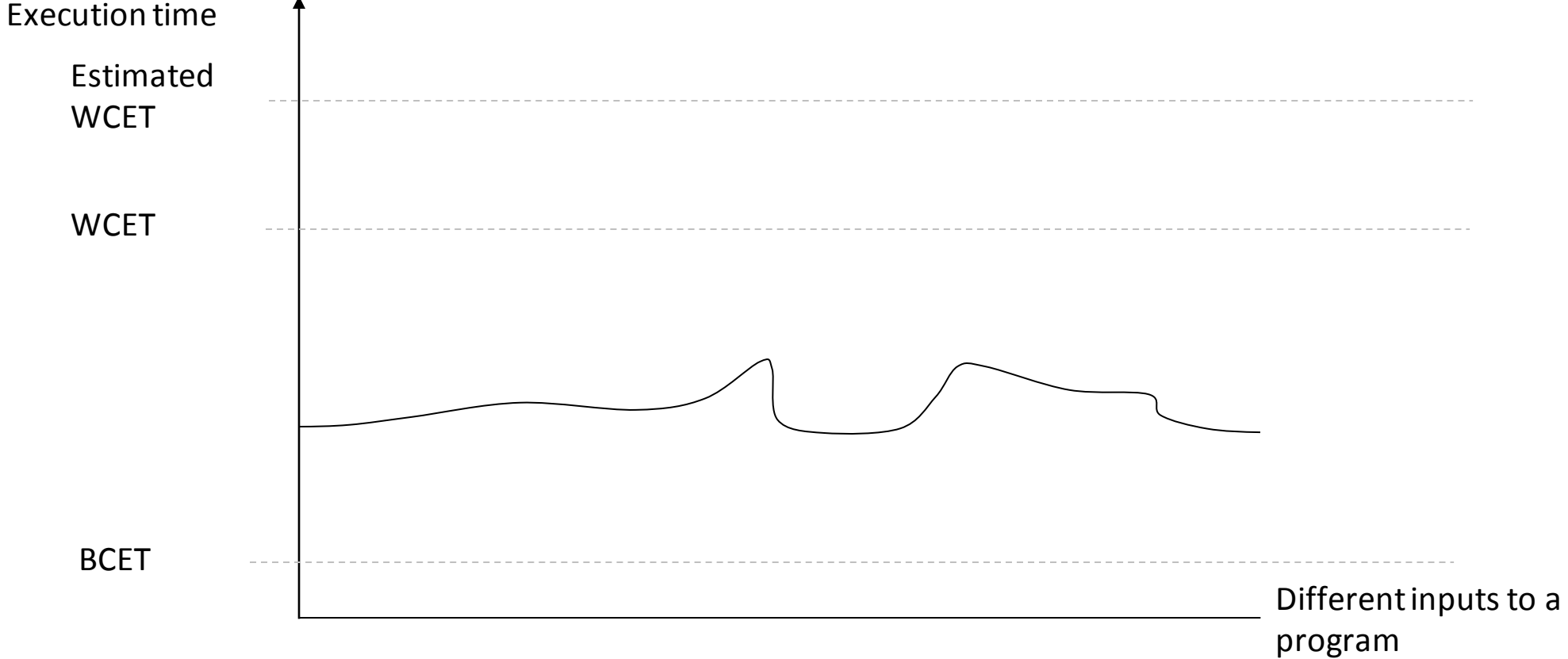
Hardware

- Hardware is often not well documented.
- Even if hardware is documented, documentation is often incorrect.
- Even if hardware is documented and correct, hardware changes.

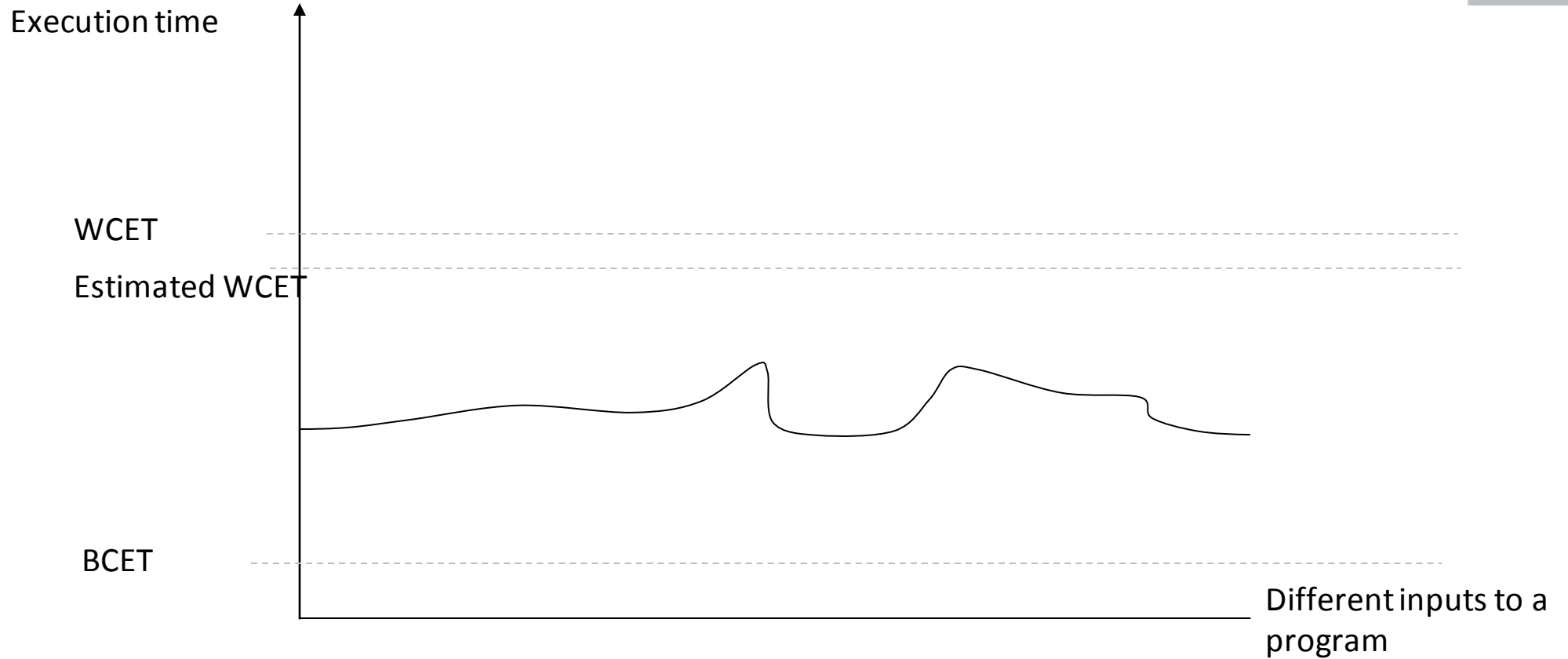


Develop
WCET analysis that does
not depend on
documentation

Ideally, we would like $WCET \leq \text{Estimated WCET}$. But on some systems, this is very hard to achieve.



It may happen in some rare cases that WCET > Estimated WCET.



The report:
Assessing the Use of Machine Learning to Find
the Worst-Case Execution Time of Avionics Software

- **Outline of this presentation**
 - Main findings
 - Background
 - Our method
 - Experimental results

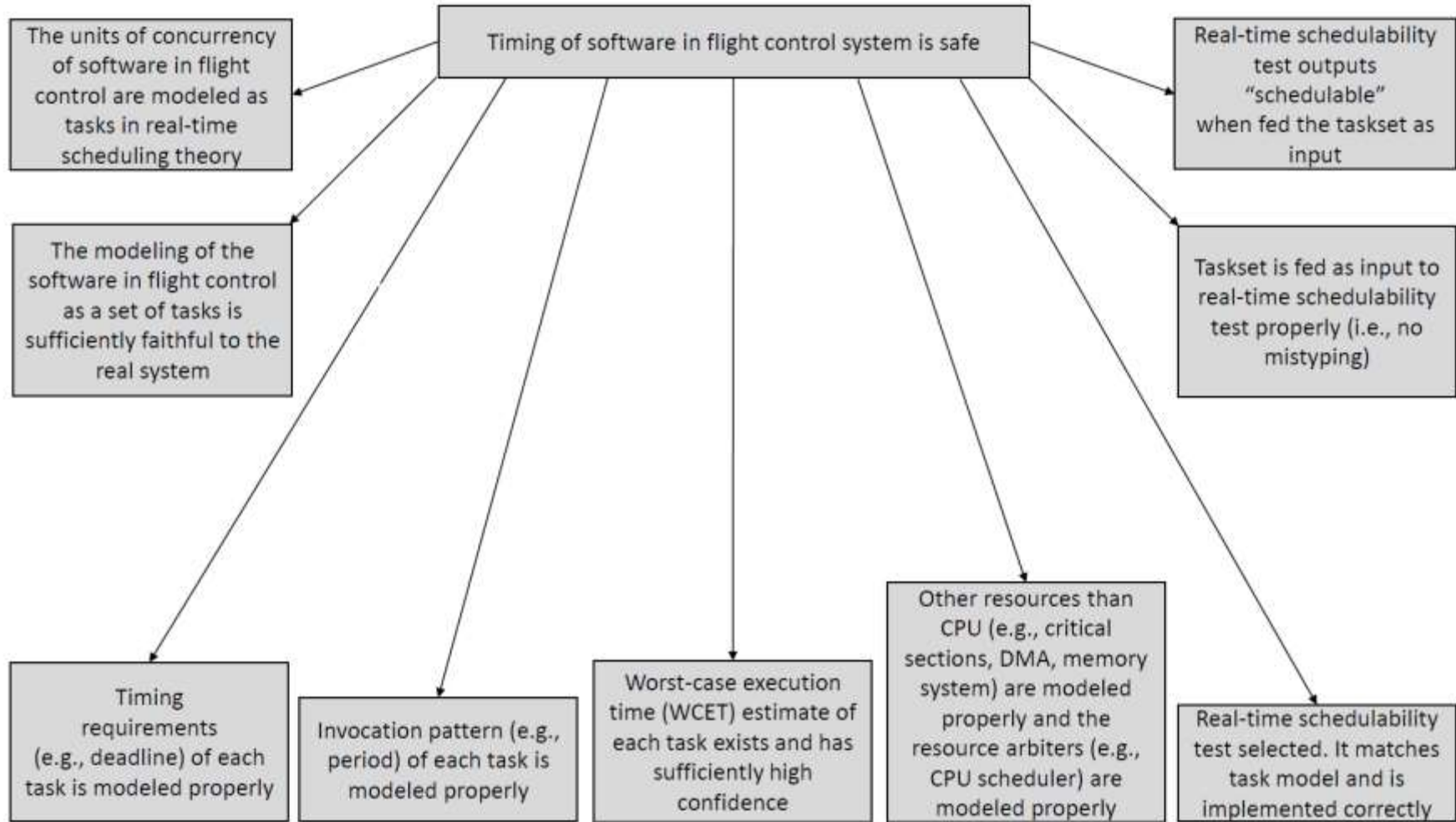
Main Findings

Findings

1. Our new WCET analysis method performs well for some target programs.
2. There are target programs for which our new WCET analysis method does not perform well.
3. Our new WCET analysis method should be viewed as a complement to other WCET analysis methods.
4. For our WCET analysis method, how to measure execution time is critical.
5. Current limitation/future work: select family of functions to learn from

Background

WCET Analysis in Assurance Case



WCET and Undocumented Hardware

Execution time of a target program depends on the hardware it executes on.

WCET analysis depends on hardware

Detailed information about hardware is typically unknown.

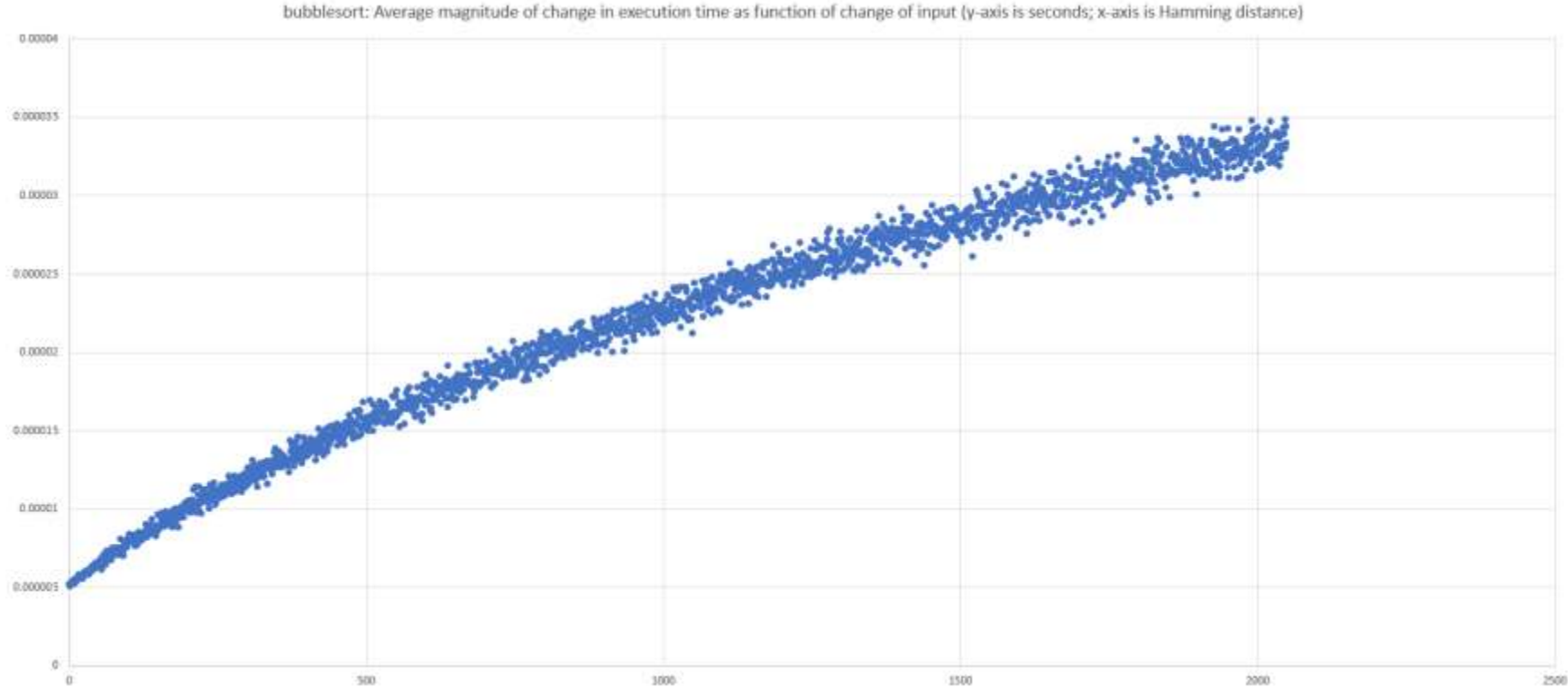
Can we perform WCET analysis without knowing the hardware?

The report provides 44 references from the research literature that points out that the lack of information about the hardware is a serious issue in WCET analysis.

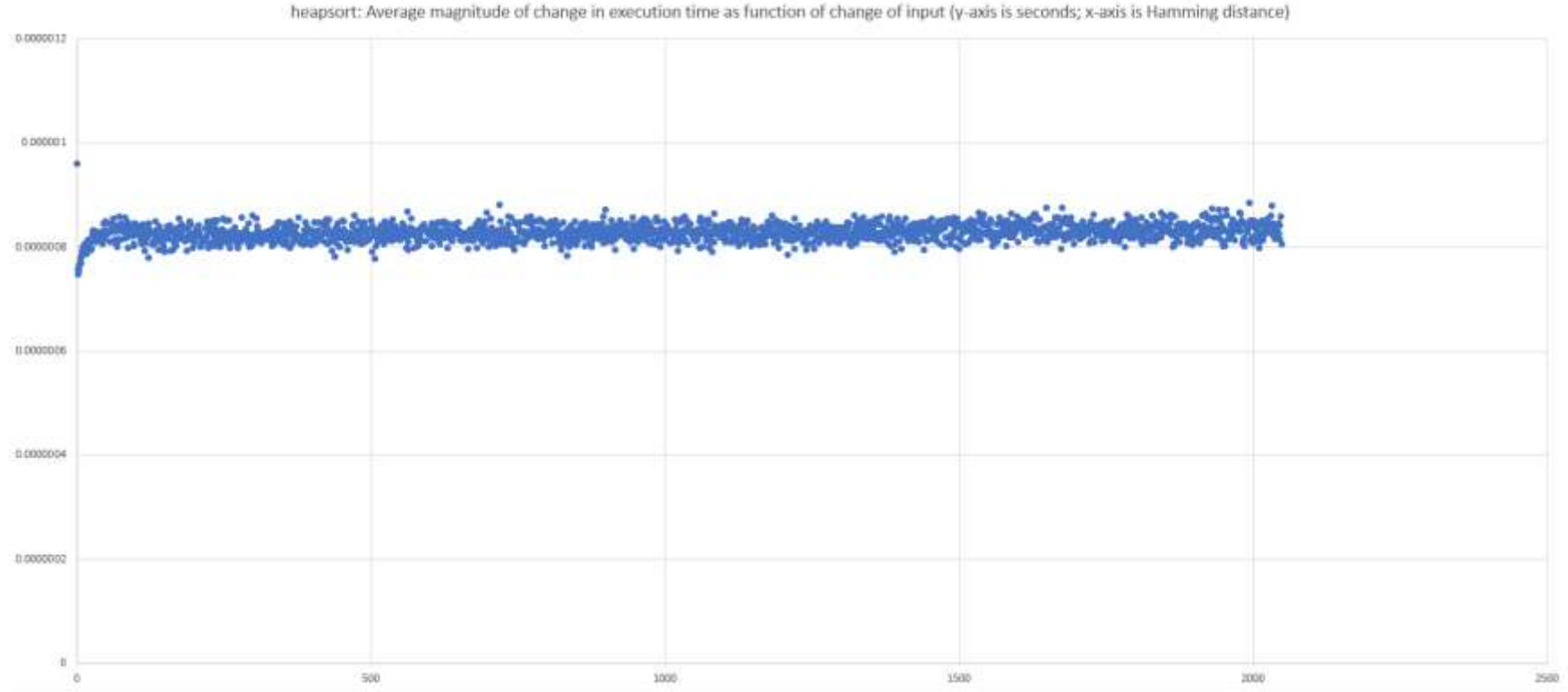
"For modern processors, technical details are often difficult to obtain due to commercial confidentiality."

Observations on Target Programs

For some target programs, the execution time as a function of input is smooth



For some target programs, the execution time as a function of input is NOT smooth



Execution time of a target program with the same input

Multiple runs of bubblesort with the same random input: Histogram of deviation of execution time from sample average of execution time (1000000 samples)



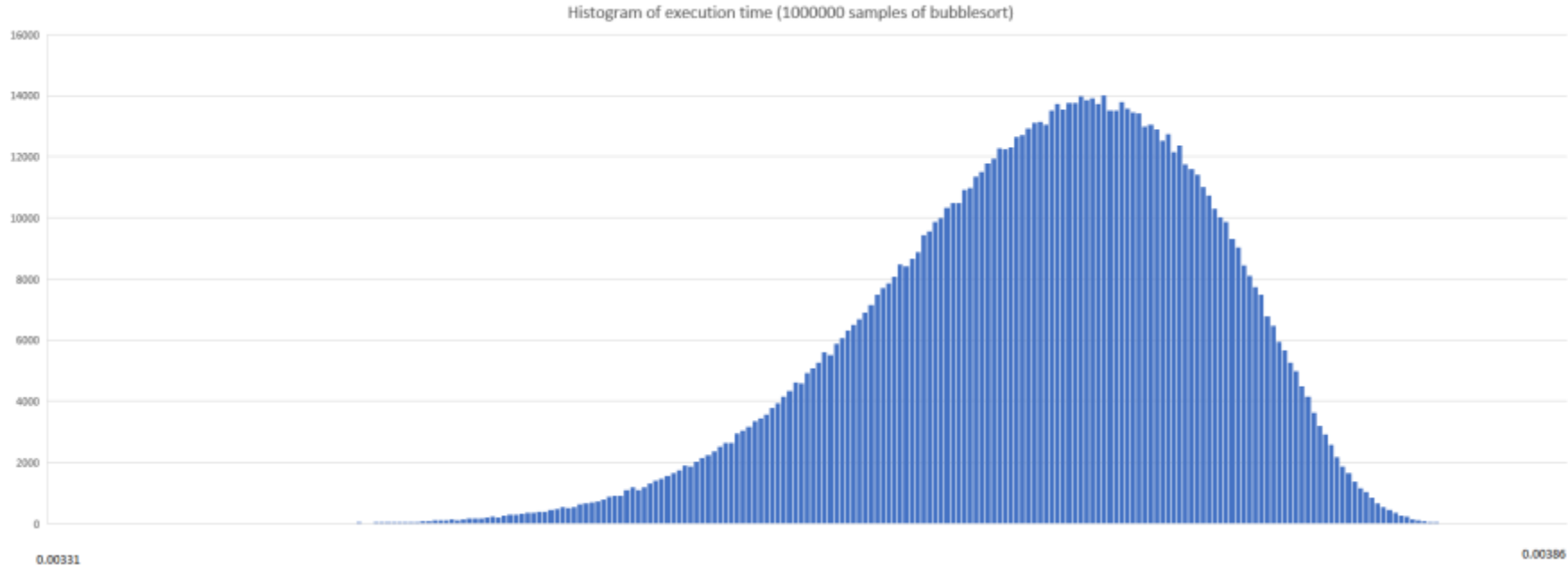
Execution time of a target program with the same input

Multiple runs of bubblesort with the same random input: Histogram of deviation of execution time from sample average of execution time (1000000 samples)

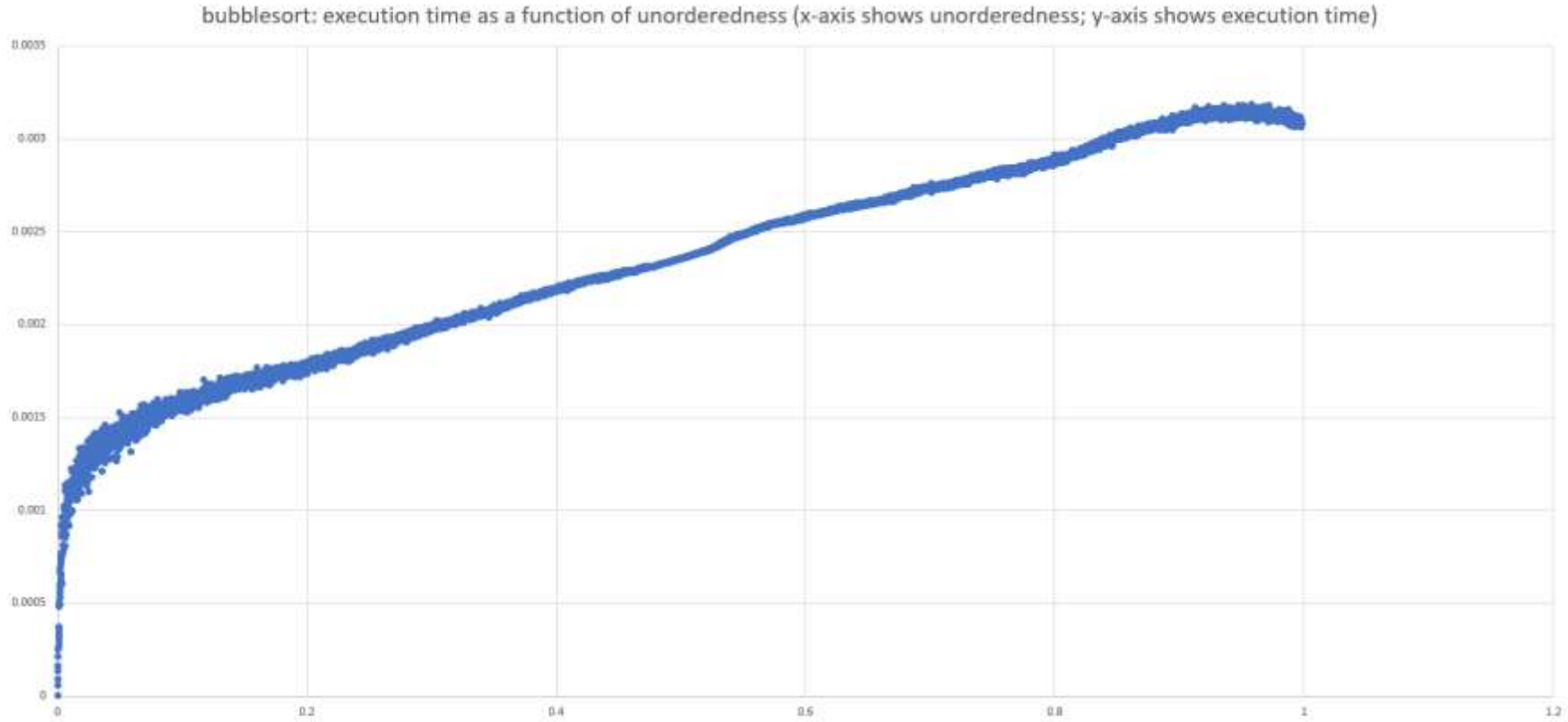


There are rare non-repeatable disturbances

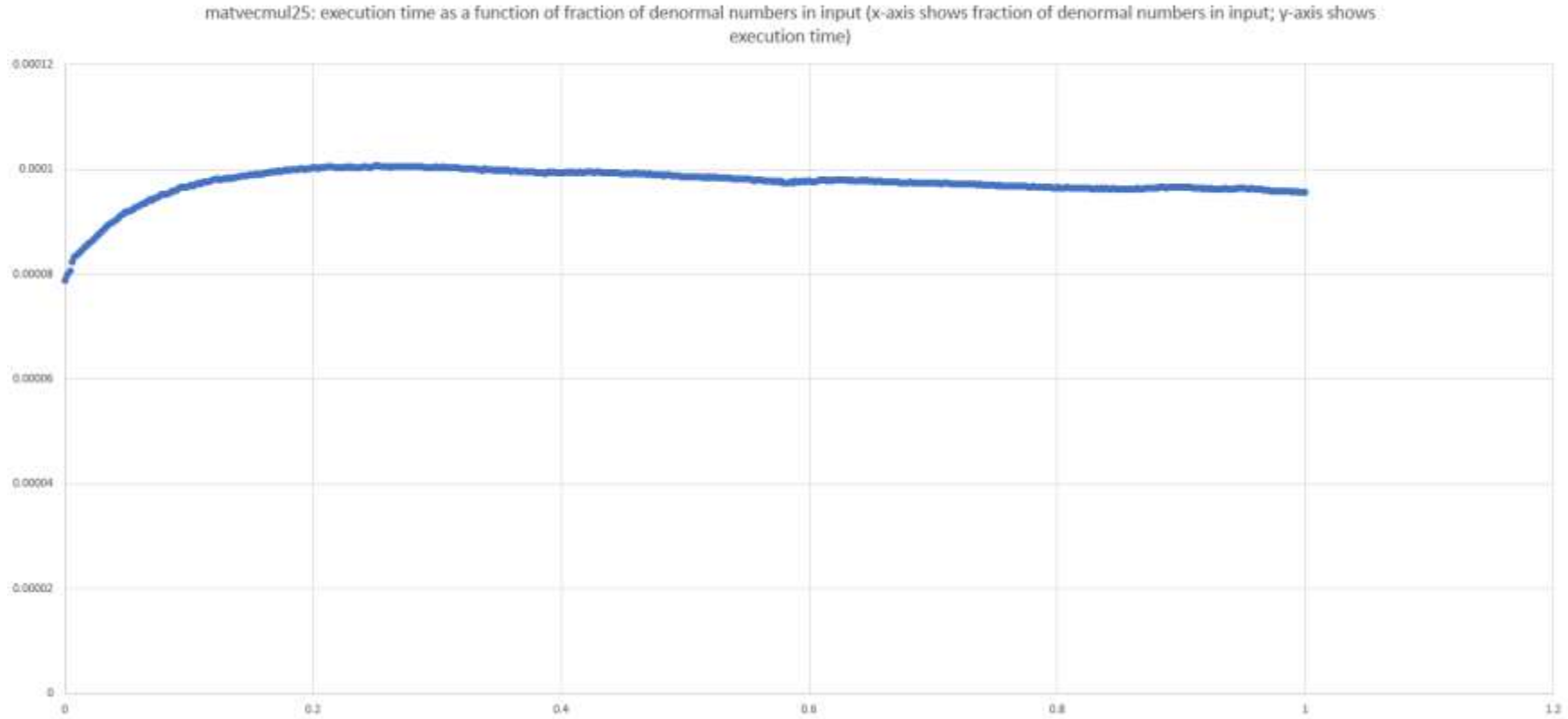
Execution time of a target program with VARYING input



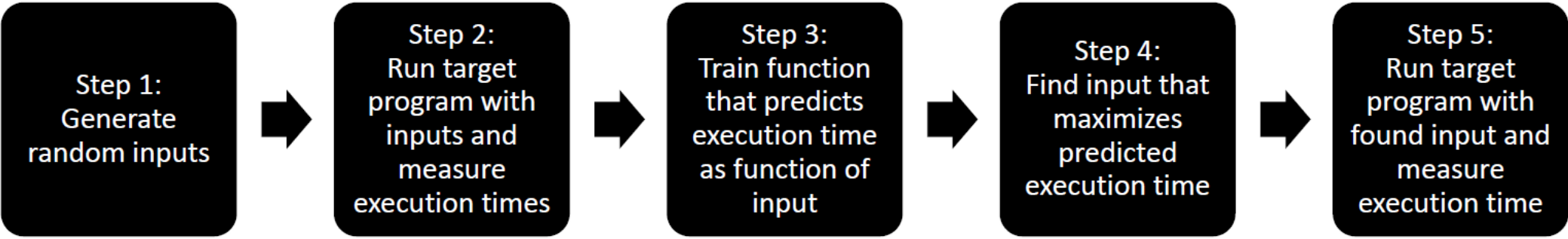
Execution time of a sorting program may depend on how sorted the input is

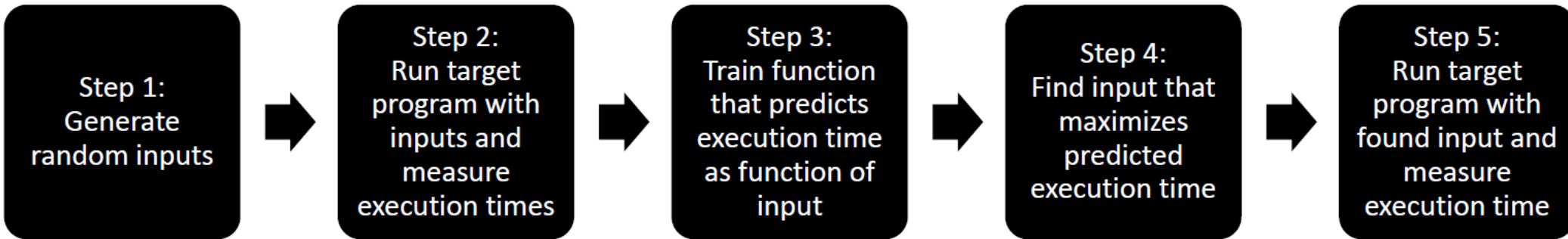


Execution time of matrix-vector multiplication depends on how many denormal numbers there are



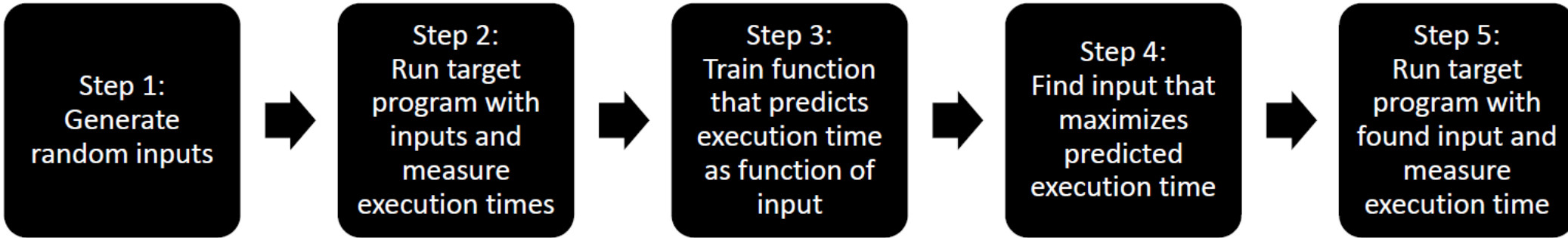
Our Method





Issues

1. Too few inputs with extreme execution times
2. Rare non-repeatable increase in execution time
3. Function learned is not sufficiently expressive
4. Training takes too long
5. Training requires too much main memory
6. Training yields a function that does not generalize
7. Training yields a function that cannot extrapolate
8. Finding the input that maximizes the predicted execution time takes too long
9. The memory mapping of Step 2 is different from the memory mapping of Step 5
10. The WCET estimation produced applies only to one memory mapping



Addressing Issues

1. Too few inputs with extreme execution times
2. Rare non-repeatable increase in execution time
3. Function learned is not sufficiently expressive
4. Training takes too long
5. Training requires too much main memory
6. Training yields a function that does not generalize
7. Training yields a function that cannot extrapolate
8. Finding the input that maximizes the predicted execution time is difficult
9. The memory mapping of Step 2 is different from Step 5
10. The WCET estimation produced applies only to one memory mapping

- ⇒ Select the extreme and train on them
- ⇒ Run each input many times (median)
- ⇒ Learn affine and then add terms
- ⇒ Solve optimization problem (loss func) approx
- ⇒ Store bits as bits (rather than FP)
- ⇒ Use fewer weights than #examples
- ⇒ Use affine function
- ⇒ Solve optimization problem approx
- ⇒ Use same MM in Step 2 as in Step 5

Evaluation

		Performance of WCET analysis			
[mint,maxt]		WCET estimate		WCET estimate / maxt	
		Only affine	With basis functions	Only affine	With basis functions
BUBBLESORT	[0.001985610, 0.002679730]	0.002707772	0.002700042	1.010464487	1.007579868
QUICKSORT	[0.000065130, 0.000081410]	0.000058219	0.000062760	0.715133276	0.770912664
QSORT	[0.000064910, 0.000073370]	0.000057520	0.000058949	0.783971651	0.803448276
HEAPSORT	[0.000088070, 0.000104480]	0.000089639	0.000090349	0.857953675	0.864749234
MERGESORT	[0.000068540, 0.000077930]	0.000063249	0.000065189	0.811612986	0.836507122
MATVECMUL25	[0.000068730, 0.000079910]	0.000086819	0.000070639	1.086459767	0.88398198
MATVECMUL700	[0.001922460, 0.002238300]	0.002380286	0.001951602	1.06343475	0.871912612
FFT_RECT	[0.000115860, 0.000136100]	0.000116229	0.000116228	0.853997061	0.853989713
FFT_POL	[0.000129760, 0.000156570]	0.000143058	0.000142638	0.913699943	0.911017436
FFT_REALONLY	[0.000271150, 0.000313680]	0.000271976	0.000271896	0.867049222	0.866794185

Good performance

Conclusion

Findings

1. Our new WCET analysis method performs well for some target programs.
2. There are target programs for which our new WCET analysis method does not perform well.
3. Our new WCET analysis method should be viewed as a complement to other WCET analysis methods.
4. For our WCET analysis method, how to measure execution time is critical.
5. Current limitation/future work: select family of functions to learn from

More information at:

“Assessing the Use of Machine Learning to Find the Worst-Case Execution Time of Avionics Software,”

DOT/FAA/TC-23/06,

Available at:

https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/research/TC2306