P vs NP

Gilberto Rodrigo Diduch

Calm Computing, Campo Largo, PR, Brazil

Summary
Cryptanalysis can take long time, is it possible to make it faster, in Turing Polynomial Time? The question about processing huge data amount in a short time remains for decades, what took us to the dilemma P vs NP, and now I am pleased to say that the search is finally completed, P is not equal to NP.

Key words:
Turing, Machines, Cryptanalysis, P vs NP and Solution.

1. Pi* algorithm

In the simple algorithm:
* Make the division of two numbers and show all the decimal places.
When the numbers in question are 355 and 113, it becomes easy to check, but not easy to find all the decimal places, because we are talking about Pi* and probably we are going to the infinite on running this algorithm.
There is not other way to make a division or to simplify it, what proves NP can't be equal to P. You will never run in polynomial time something that can go to infinite; But we don't even need to force the infinite. Just see the algorithm:
* Make the division of two numbers and show 5.000.000.000.000 with 17 exponent, decimal places. (Not eccentric, Shigeru Kondo and Alexander Yee got near this number on the year 2010, nowadays, and Pi* magics human thoughts since 1650 B.C., registered by the Egyptian on the Rhind's papyrus). or
* Make the division of two numbers and keep showing decimal places within a time equal to the time spent on any known exponential time algorithm ofyour choice.
The matter isn't how you ask
The matter is what you ask
If you ask huge data process, it will take time for sure.

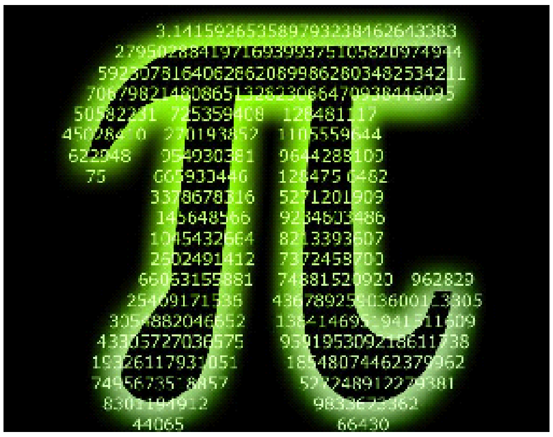

Figure 1.

2. Wood example

If you want to make a wood house, you will have to process the wood, obviously, now, if you want to cover all the Earth surface with wood (imagine that it were possible, someone invented a super genetic modified specie of three that get grown at the same time it takes you to process the wood, then you would have infinite source of wood), you still would have to process the wood same way, same to some cryptanalysis, the distance run by the hard disk reading, or the distance that electricity runs inside the computer could paint all the Earth surface not just once, because you need to reach the information, many data to compare, many possibilities to check; and this distances, even small, when they are put all together, it results in a huge distance, that takes time to be run.
If you want the wood cut, you will have to cut it, no magic can be done.
If you want data processed, you will have to process it.

Manuscript received January 5, 2012
Manuscript revised January 20, 2012

Figure 2.

Figure 3.

Figure 4.

3. Philosophizing

Talking about Pi*, is it possible that any remainder of the division 355 by 113 won't be equal to 113 or 226 or the multiple, among supposed infinite remainders? It would be the last decimal place on the division. Same to the question 'Are we alone in the Universe?'.
Taking the opportunity, I would like to let some thoughts, mathematics philosophy time, all things considered infinite have to be on constant growing or on a cycle, and they are finite on the exactly step they are on their grow or cycle, what can't be proven for the Universe size, or the size of the huge all; to imagine Universe is infinite is same unreasonable as saying it is finite, it is too hard to comprehend some with no ending, but so it is to think that it have an end; and what have one meter before the imagined end? And if the Universe keep growing or expanding, what does have one meter before Universe last atom or last meter of vacuum? Or like absence of space or an non atomic system, something hard even to imagine.

4.Coming back to P vs PN

Even if you make atomic computers or nano computers, speeding much the time of processing data, P will still be P and NP will still be NP, different, as their names suggest.
Trying to do an exponential time program run in the Turing polynomial time is same as looking for the gold pot on final of the rainbow.

Figure 5.

5. New Machine

Taking the opportunity once again, I would like to show a new theoretical machine, continuing the knowledge left by Turing.
It's based on the possibility of 'writing' the 0s and 1s on the atoms, instead of having the spaces in a type, all the Turing spaces will be set inside of just one space, and the reader

won't need to move through the spaces, it will be kept fixed on the unique space, no distances to run, no time wasting. When you use atoms instead of spaces on a type you do good use of space and can safe much time.

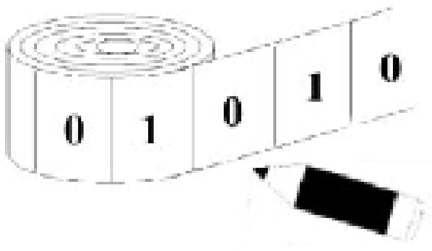

Figure 6.

Figure 7.

Figure 8.

And Bill, you can phone call me, I have a strong educated guess on how to make the atom inscription, but it is not subject for this lines.
Even speeding up the time of processing, like in 1 billion times, you will be speeding either P and NP, their difference will remain, just speeded up.

6. Finishing

You have just two possibilities on trying to prove that P is equal to NP, first is on the construction of the algorithms, the Pi* algorithm proves they are different; and second possibility is on the machine structure, speeding it up; once it will speed up both, P and NP, their difference, proven with Pi* algorithm, will remain in any speed of the computer; then we have the two negative necessary to make sure that P is not equal to NP.
Pi* for P (vs NP), some coincidences are really great, if they exist, coincidences.
The answer for P vs NP is a categorical NO; they are different as their names suggest.

Figure 9.