A new methodology to tackle the P versus NP problem

Mario J. Pérez-Jiménez

Research Group on Natural Computing
Dpt. Computer Science and Artificial Intelligence
University of Seville, Spain
Academia Europaea (The Academy of Europe)

www.cs.us.es/~marper marper@us.es

17th Brainstorming Week on Membrane Computing
Sevilla, Spain, February 5, 2019

Computers ...

as a singular ...

Limits to what computers can do?

The P versus NP problem

- ∗ **Finding** solutions versus **checking the correctness** of solutions.
- ∗ **Proofs** versus **verifying their correctness**.

- **Central problem** of Computer Science.

The P versus NP problem

It is widely believed that it is harder

* **to solve** a problem than **to check** the correctness of a solution

It is widely believed that $\mathbf{P} \neq \mathbf{NP}$.

P: class of problems which can be “quickly” solved.

Attacking the P versus NP problem

NP-complete problems: hardest in the class **NP**.

Classical approach (1970):

- **P $\neq$ NP**.
  - ∗ Find an **NP**-complete problem such that it does not belong to the class **P**.
- **P $=$ NP**.
  - ∗ Find an **NP**-complete problem such that it belongs to the class **P**.

Tractability versus intractability

Tractable problem with regard to a complexity measure:

- It can be solved by a DTM using a polynomial amount of resources.
- The upper bound of the computational resources is polynomial.

P: class of decision tractable problems.

Intractable problem with regard to a complexity measure:

- The lower bound of the computational resources is exponential.
- There exist intractable problems with regard to any complexity measure.

Tractability versus intractability

NP-complete problems are considered **presumably intractable** problems.

$(\mathbf{P} \neq \mathbf{NP}) \Longleftrightarrow$ (Any **NP**-complete problem is intractable with regard to the time)

$(\mathbf{P} = \mathbf{NP}) \Longleftrightarrow$ (Any **NP**-complete problem is tractable with regard to the time)

Computing models

Computing model: A mathematical theory.

⋆ *Resolution* of an abstract problem by means of a *mechanical procedure*.

Efficiency of **computing models**:

⋆ Ability to provide polynomial-time solutions to intractable problems.

Presumed efficiency of **computing models**:

⋆ Ability to provide polynomial-time solutions to **NP**-complete problems.

RGNC

Efficiency and presumed efficiency

Non-efficient computing models: only problems in **P** can be solved in poly-time.

The model of **DTM**s is non-efficient.

The model of **NDTM**s is presumably efficient.

If **P** $\neq$ **NP** then:

- ⋆ The model of **NDTM**s is efficient.
- ⋆ Any presumably efficient computing model is an efficient one.
- ⋆ A computing model can be neither efficient nor presumably efficient (**Ladner theorem**).

Extension of a computing model

Given two computing models M_1 and M_2:

- ⋆ M_2 is an **extension** of M_1 if every mechanical procedure of M_1 is also a mechanical procedure of M_2.

If M_2 is an *extension* of M_1 then M_2 can be obtained from M_1 by **adding** some syntactic or semantic **ingredients**.

Frontiers of the efficiency

Let M_1 and M_2 be two computing models such that:

(a) M_1 is non-efficient.

(b) M_2 is an *extension* of M_1

(c) M_2 is presumably efficient.

Passing from M_1 to M_2:

- ⋆ Passing from non efficiency to presumed efficiency.
- ⋆ Provides a frontier between tractability of abstract problems and the presumed intractability.

Frontiers of the efficiency

Non efficiency

M_1

Presumed efficiency

M_2

Efficiency and presumed efficiency

A new methology to tackle the **P** *versus* **NP** problem:

- $\boxed{\mathbf{P} = \mathbf{NP}}$: the ingredients added to obtain M_2 from M_1 do not play a relevant role to obtain efficient solutions to **NP**-complete problems in M_2.

- $\boxed{\mathbf{P} \neq \mathbf{NP}}$: the ingredients added to obtain M_2 from M_1 are crucial to obtain efficient solutions to **NP**-complete problems in M_2.

Efficiency and presumed efficiency of membrane systems

Let $\mathcal{R}$ be a class of recognizer membrane systems.

* $\mathcal{R}$ is non-efficient if and only if $\mathbf{P} = \mathbf{PMC}_{\mathcal{R}}$.
* $\mathcal{R}$ is presumably efficient if and only if $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{R}}$.

Frontiers: Cell-like with active membranes

Non − Efficiency	Presumed Efficiency	
NAM	**AM**	*(adding rules)*
$\mathbf{AM}^0(-d,+ne)$	$\mathbf{AM}^0(+d,+ne)$	*(adding rules)*
$\mathbf{AM}^0(-d,+ne)$	$\mathbf{AM}(-d,+ne)$	*(polarization)*

$\mathbf{AM}^0(+d,-ne)$?

Frontiers: Cell-like with symport/antiport rules

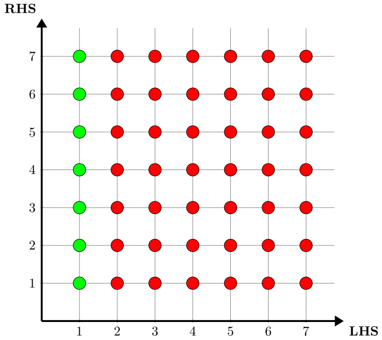
Non – Efficiency	Presumed Efficiency	
CDC(1)	**CDC**(2)	(*length*)
CSC(2)	**CSC**(3)	(*length*)
CSC(2)	**CDC**(2)	(*kind*)
$\widehat{\textbf{CSC}}(2)$	$\widehat{\textbf{CDC}}(2)$	(*kind*)
$\widehat{\textbf{CSC}}$	**CSC**	(*environment*)

Frontiers: Tissue-like with symport/antiport rules

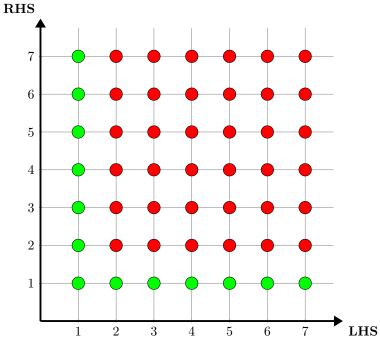
Non − Efficiency	Presumed Efficiency	
TC	**TDC**	(*adding rules*)
TDC(1)	**TDC**(2)	(*length*)
TDA(1)	**TDA**(3)	(*length*)
TDS(1)	**TDS**(3)	(*length*)
TC	**TSC**	(*adding rules*)
TSS	**TSC**	(*direction*)
TSS	**TSA**	(*direction*)
TSS(3)	**TSA**(3)	(*direction*)
TSS(2)	**TSS**(3)	(*length*)
TSA(2)	**TSA**(3)	(*length*)
$\widehat{\mathbf{TSC}}$	**TSC**	(*environment*)
$\widehat{\mathbf{TSC}}(3)$	**TSC**(3)	(*environment*)
TSC(2)	**TDC**(2)	(*kind of rules*)
$\widehat{\mathbf{TSC}}$	$\widehat{\mathbf{TDC}}$	(*kind of rules*)
$\widehat{\mathbf{TSC}}(k), k \geq 2$	$\widehat{\mathbf{TDC}}(k), k \geq 2$	(*kind of rules*)

Frontiers: Tissue-like with evolutional communication rules

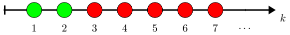
$\mathbf{PMC_{TDEC(k_1,k_2)}}$



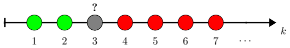
$\mathbf{PMC_{TSEC(k_1,k_2)}}$



$\mathbf{PMC_{TDEC(k)}}$



$\mathbf{PMC_{TSEC(k)}}$



$\mathbf{P} = \mathbf{PMC}_{\mathcal{R}}$ $\mathbf{DP} \subseteq \mathbf{PMC}_{\mathcal{R}}$

THANK YOU

FOR YOUR ATTENTION!