PV270 Biocomputing

Complexity Issues

D. Safranek, 2023

What we miss

- A sharp understanding of biochemical laws controlling life
- Suitable models for nano-level computing, other than the “rigid” Turing machine model

Molecular Computing – perspectives

- Over-optimism, over-pessimism
- What can we compute with DNA ?
  - “Killer” application is needed – challenge for computer scientists
  - Better algorithms than exhaustive search – same comment
  - We need better biotech tools to control the molecules (do they exist already?) – challenge for biotech
  - Cope with the errors: impact on the size of the solutions (in number of strands)
  - How much can we compute – SAT up to 70-80 variables

    => impact on the size of the solutions (in number of strands)

Molecular Computing – perspectives

Positive side

- Applications to biotechnology: e.g., a SAT implementation used to execute Boolean queries on a “wet” database, based on some tags (IDs)

- Useful in specialized environments: e.g., extreme energy efficiency or extreme information density required

- Provide the means to control biochemical systems just like electronic computers provide the means to control electromechanical systems

Molecular Computing – perspectives

Negative side

- At this moment, we cannot control the molecules with the precision the physicists and electrical engineers control electrons
- Need of a breakthrough in biotechnology: more automation, more precise techniques
- Example:
    - HPP may be solved nowadays on electronic computers for graphs with 13 500 nodes
    - Adleman's approach scaled up for graphs with 200 nodes needs more DNA than the weight of the Universe

Molecular Computing – perspectives

Killer Application Wanted

- Some inspirations in parallel computing (NC — Nick’s complexity class):
  - Polylogarithmic runtimes
  - Polynomial numbers of cores
- DNA computing challenge:
  - Achieve polylogarithmic runtime with polynomially bounded volumes of DNA

Complexity Issues

Weak Model

- The instruction set acting on multi-sets in constant time:
  - remove(U,{S1,…,Sn})
  - union({U1,…,Un}, U)
  - copy(U,{U1,…,Un})
  - select(U)

Complexity Issues

Weak Model

- Time complexity in number of biological steps:
  - Can we do each of the parallel operations as a single biological step?
  - pour(U,U’): make a union (U := U ++ U’)
  - In fact, union can require n pour operations if we serialise the “chemical” instrumentation…

Complexity Issues

Strong Model

- remove(U,{S1,…,Sn})
  - label the strings S1,…,Sn in U by oligonucleotide sequences
  - mix with restriction enzymes to “cut” the strings out
  - Cannot be done in parallel!
  - So we have O(n)

Complexity Issues

Strong Model

- union({U1,…,Un}, U)
    - For each i do:
        - pour(U, Ui)
    - So we have O(n)

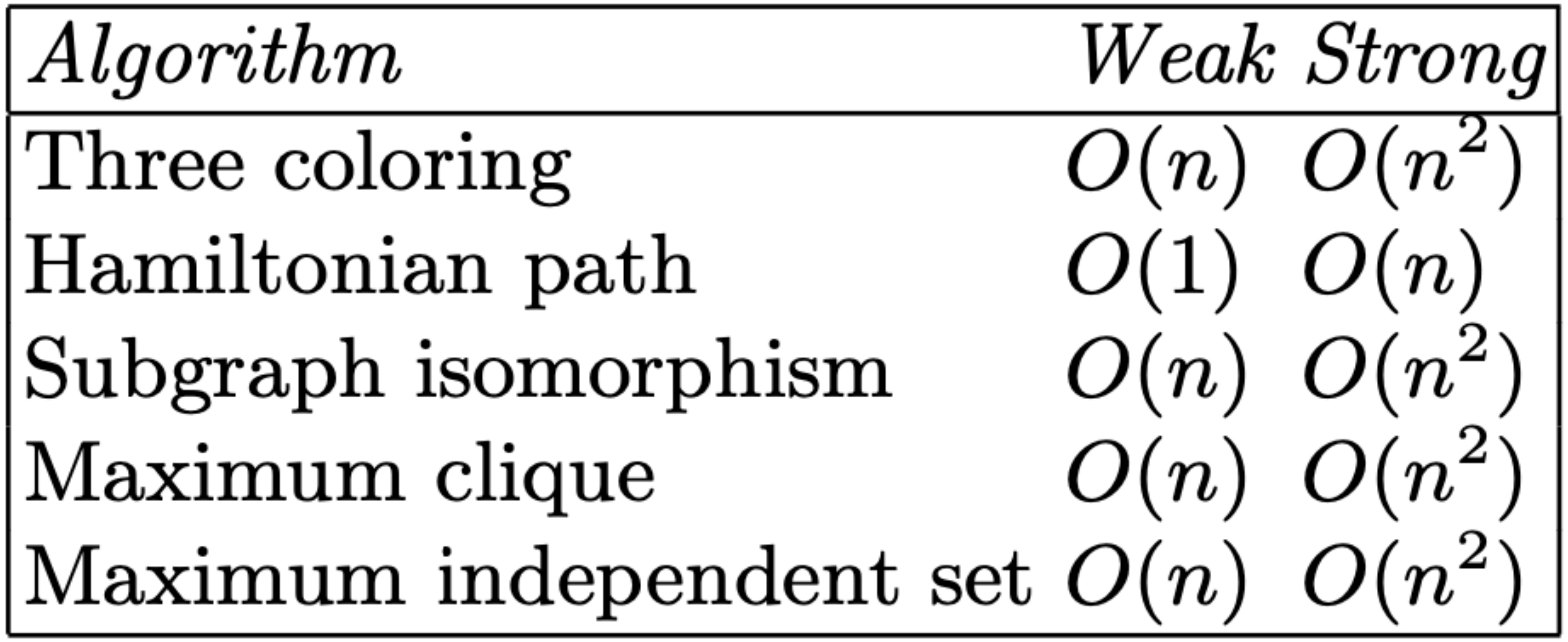
Complexity Issues

Strong Model

- copy(U,{U1,…,Un})
  - For each i do:
    - duplicate every string in U and put the duplicates in a separate tube (e.g. use one iteration of PCR)
  - So we have O(n)

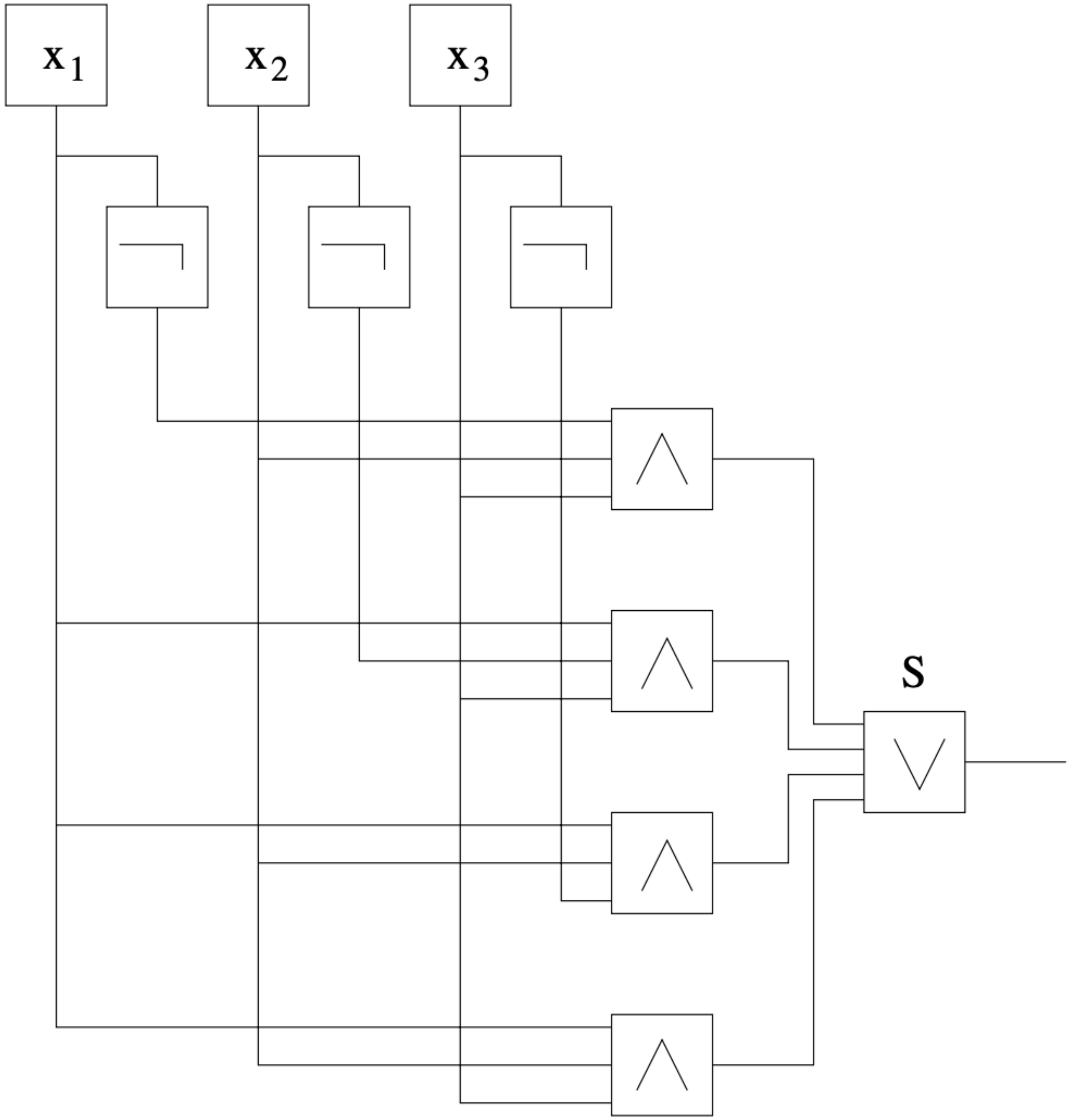
Complexity Issues

Weak vs. Strong Model



Algorithm	*Weak*	*Strong*
Three coloring	$O(n)$	$O(n^2)$
Hamiltonian path	$O(1)$	$O(n)$
Subgraph isomorphism	$O(n)$	$O(n^2)$
Maximum clique	$O(n)$	$O(n^2)$
Maximum independent set	$O(n)$	$O(n^2)$

Boolean Circuits Model



size 8

depth 3

Boolean Circuits Model

- Ogihara and Ray have simulated (physically) Boolean Circuits in DNA (using *pour* operations) in a way reflecting the structure of the given concrete Boolean Circuit

- This is an example showing the Turing power of DNA computing (Boolean Circuits are Turing complete)