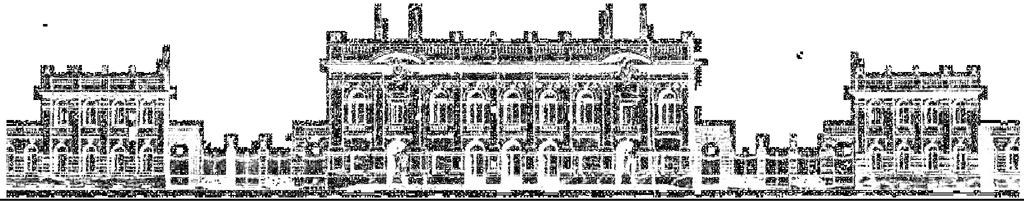


Laboratorio di Analisi di Segnale e Analisi di Rischio (LASAR)
Dipartimento Ingegneria Nucleare,
Politecnico di Milano, ITALIA

Advanced Computational Methods for the Assessment and Optimization of Network Systems

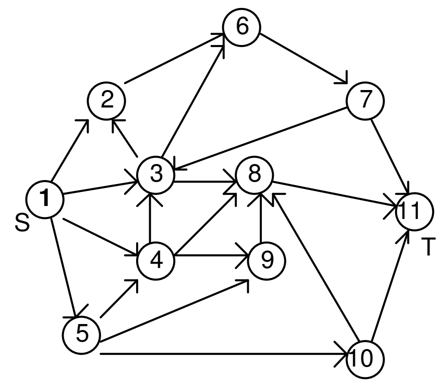
Enrico Zio
Giovanni Sansavini
Dipartimento di Ingegneria Nucleare, Politecnico di Milano

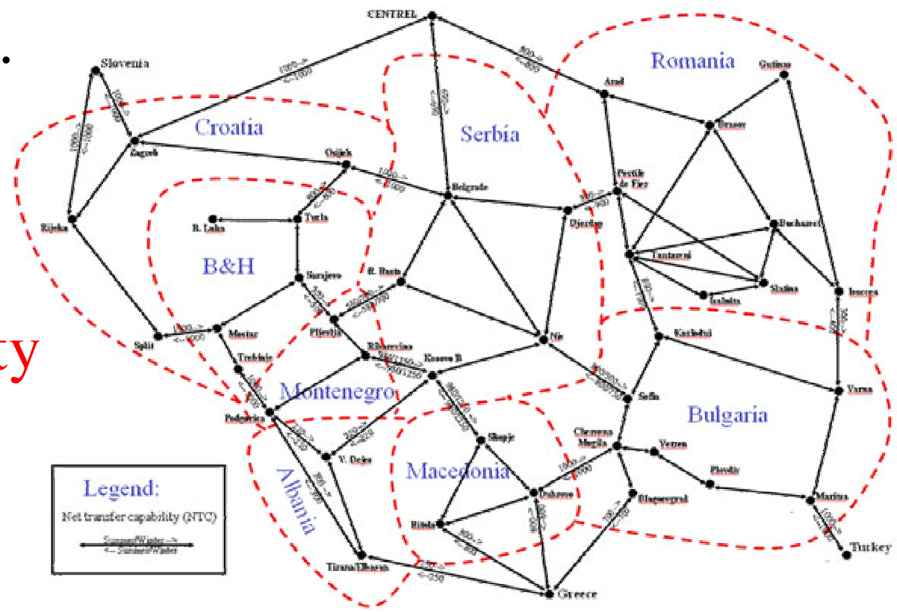
PROBLEM STATEMENT

- **WHAT?**

✓ modern society distributed services (communication, power, transportation, …)

✓ increasing reliance on technological network systems.





⇨ attention to reliability, availability, vulnerability, safety

⇨ increased system complexity

⇨ analysis by classical modelling tools difficult

NETWORK RELIABILITY

- **WHAT?**

✓ Network reliability: source-target connectivity

cut(path) sets procedure ⇨ NP-hard

- **HOW?**

✓ Cellular Automata (CA)

✓ Monte Carlo sampling and simulation (MC)

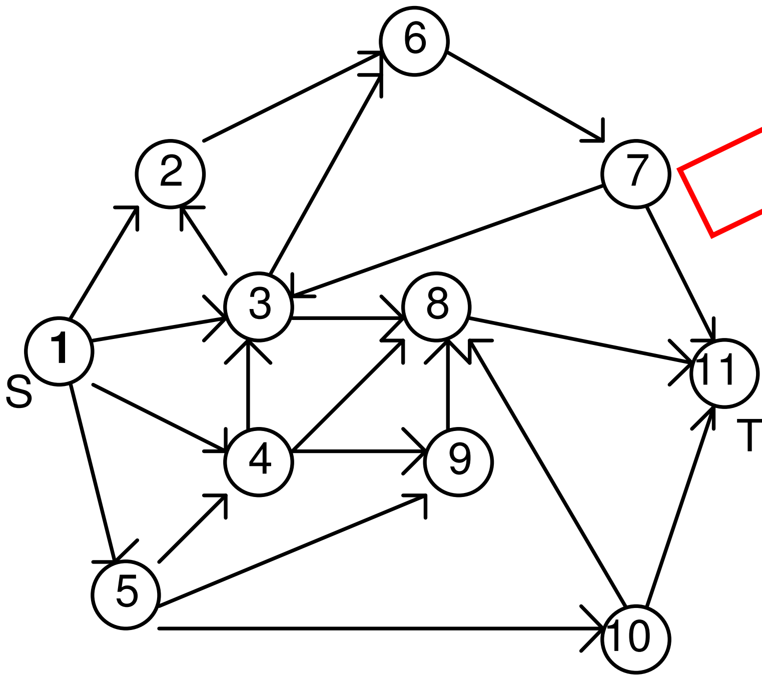
OUTLINE

- ✓ Application to network connectivity
- ✓ Application to network reliability
- ✓ Basics of MC simulation
- ✓ Application to network availability

Application to network connectivity

Application: S-T connectedness problem

(Rocco and Moreno, 2002)



1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	1	1	0	1	0	0

Node i = cell

N_i = set of cells which provide input to i

(e.g. $N_4 = \{1, 4, 5\}$)

s_i = 1 when node i is operating (active)

= 0 when not operating (passive)

$$s_i(t+1) = s_p(t) \vee s_q(t) \vee \ldots \vee s_r(t) \qquad p, q, \ldots, r \in N_i$$

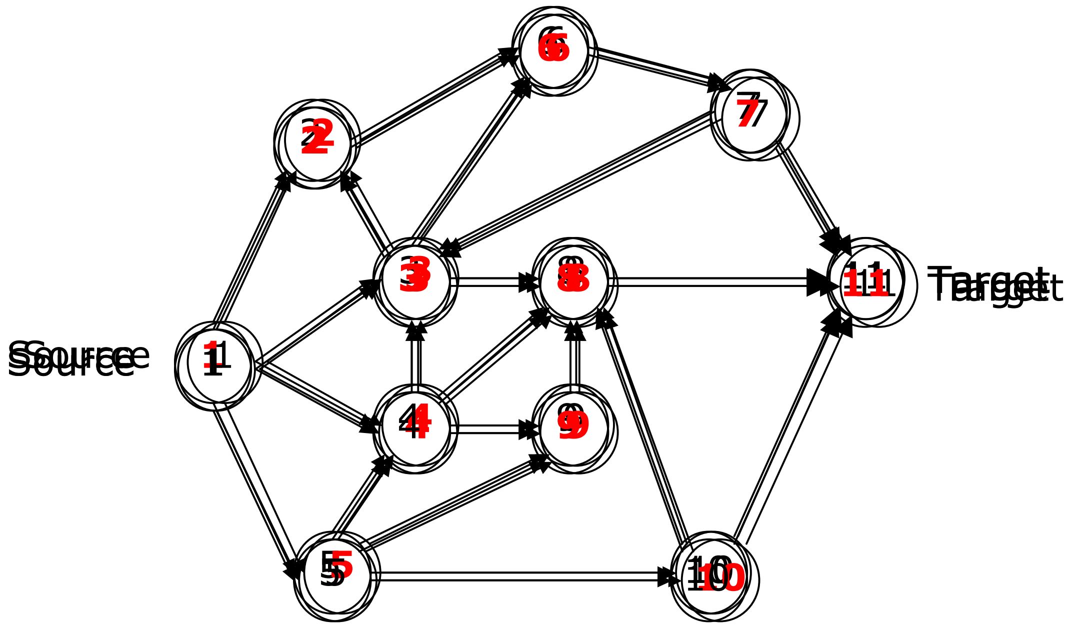
a cell is activated if there is at least one active cell in its neighbourhood

Application: S-T connectedness problem

BASIC ALGORITHM

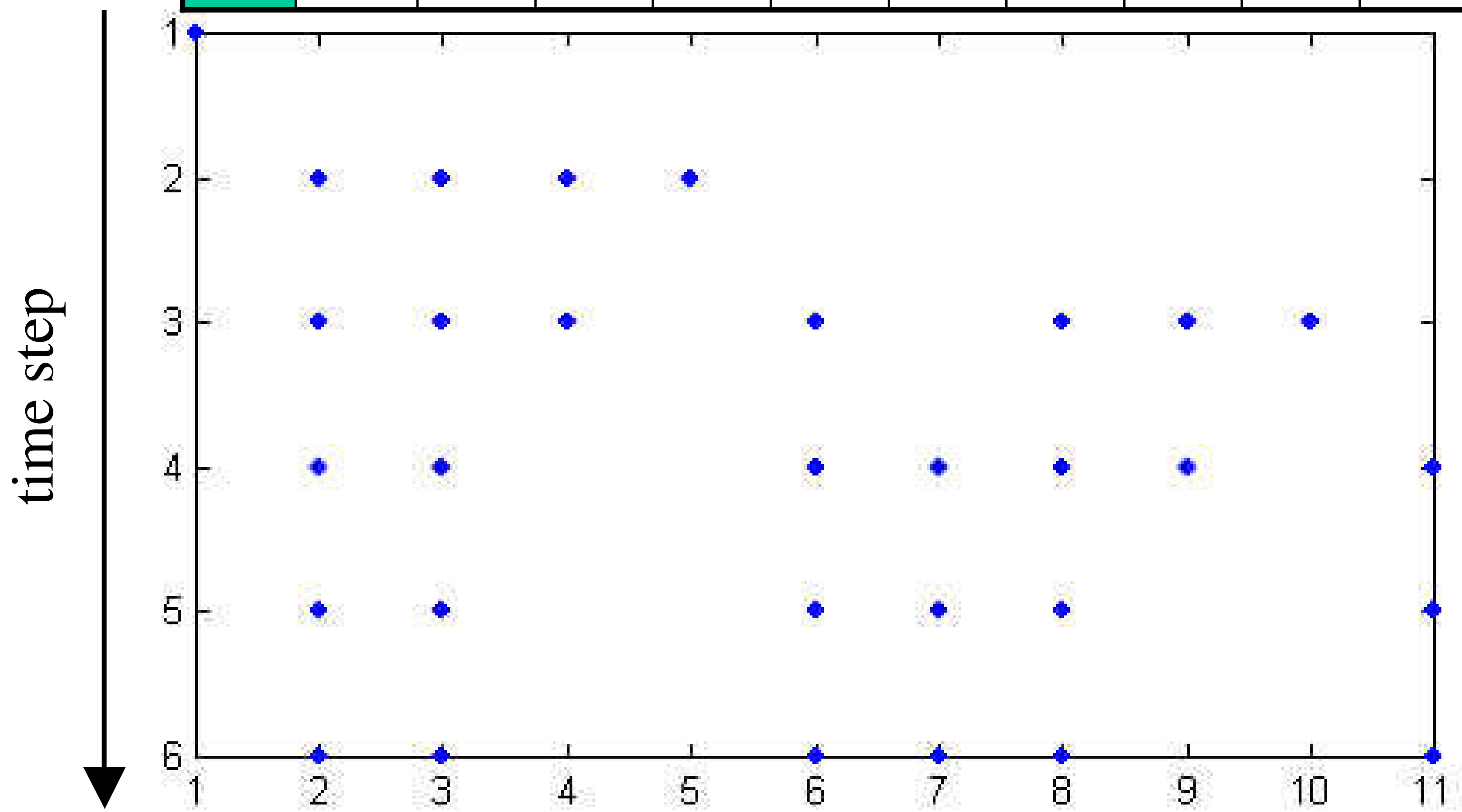
1. $t = 0$
2. Set all the cells state values to 0 (passive)
3. Set $s_s(0) = 1$ (source activated)
4. $t = t + 1$
5. Update all cells states by means of CA rule
6. If $s_T(t) = 1$, stop (target activated), else
7. If $t < m - 1$, go to 4. Else
8. $s_T(m-1) = 0$ (target passive): no connection path from S to T

Application: S-T network evolution



Application: CA evolution

1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0



Application to network reliability

Application: S-T network reliability

(Rocco and Moreno, 2002)

w_{ji} = 1, success $\rightarrow$ p_{ji}

= 0, failure $\rightarrow$ 1-p_{ij}

$$s_i(t+1) = [s_p(t) \wedge w_{pi}] \vee [s_q(t) \wedge w_{qi}] \vee \ldots \vee [s_r(t) \wedge w_{ri}]$$

$$p, q, \ldots, r \in N_i$$

a cell is activated if there is at least one active cell in its neighbourhood and the connecting arc is functioning

Application: S-T network reliability

BASIC ALGORITHM

1. $n = 0$

2. $n = n + 1$

3. Sample by MC a realization of the states of the connecting arcs ***w***

4. Apply the previously illustrated CA algorithm for *S*-*T* connectedness, to evaluate if there is a path from *S* to *T*

5. If a path exists, then update the counter of successful system states

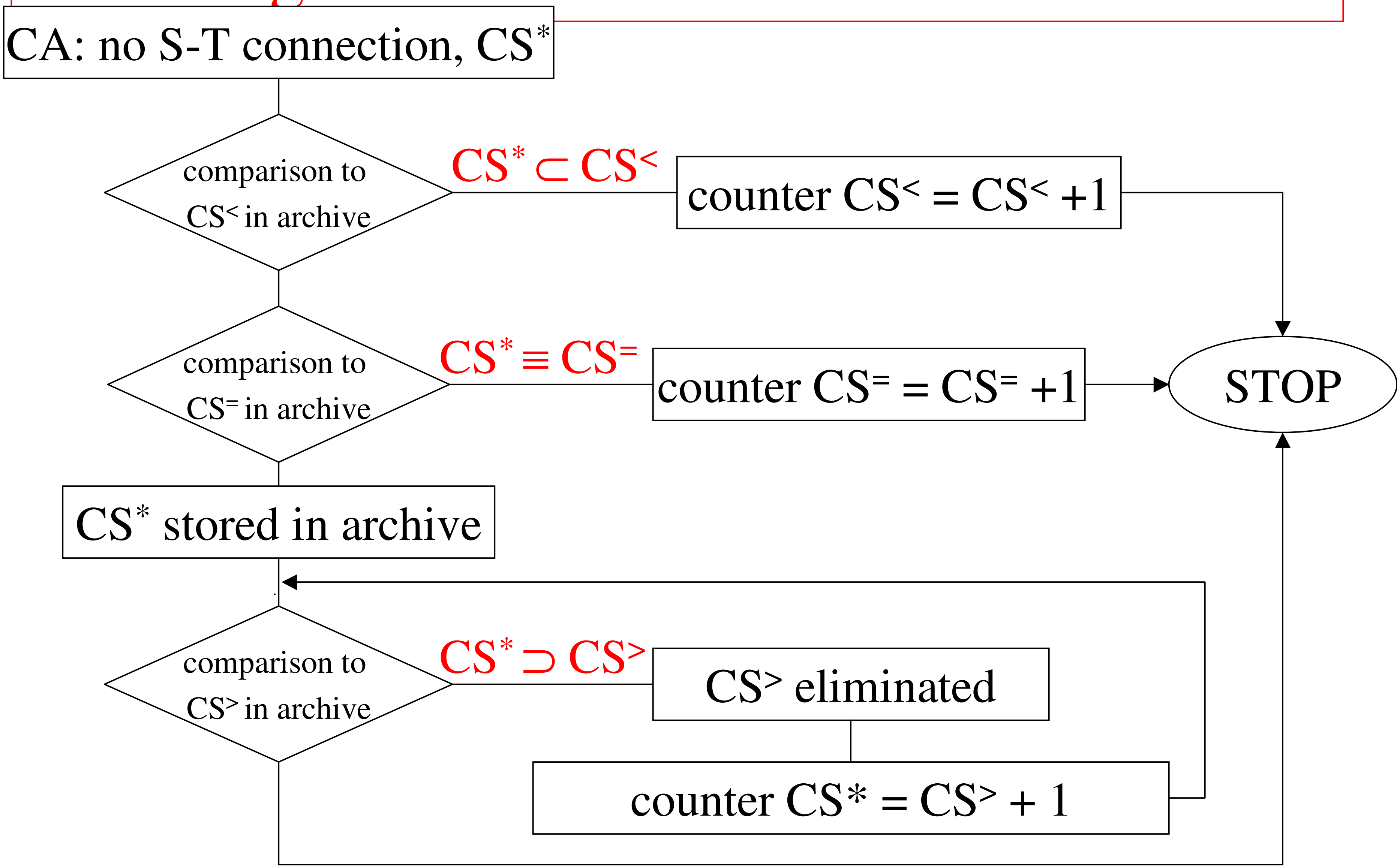
6. If MC – iteration n < N, go to 2. Else

7. Network reliability = S-T successful paths/N

Identification of minimal cut sets by CA

(Zio, Librizzi and Sansavini, 2006)

Algorithm for MCS identification



Algorithm for MCS identification

1. Keep an archive of CS

2. When a CS^* configuration is sampled
   - check whether another verified CS of $\leq$ order $\subset CS^*$ is already stored; if not
   - store the CS^* and check if higher order $\supset CS^*$

3. Update counter of occurrences for CS probability estimate

CA + MC for component importance measures

(Zio, Librizzi and Sansavini, 2006)

Algorithm for importance measure

4. Update counter of occurrences of CS containing *ji*

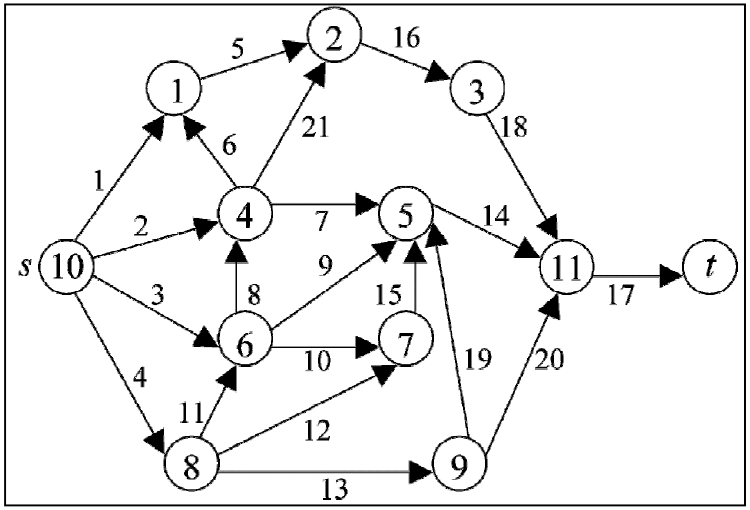
Fussell-Vesely importance measure I_{ji}^{FV}

$$I_{ji}^{FV} \approx \frac{number\ of\ occurred\ CS\ containing\ \ ji}{number\ of\ MC\ trials}$$

contribute of arc *ji* to system failure in terms of partecipation to MCS

Case study

Literature case study: MCS identification

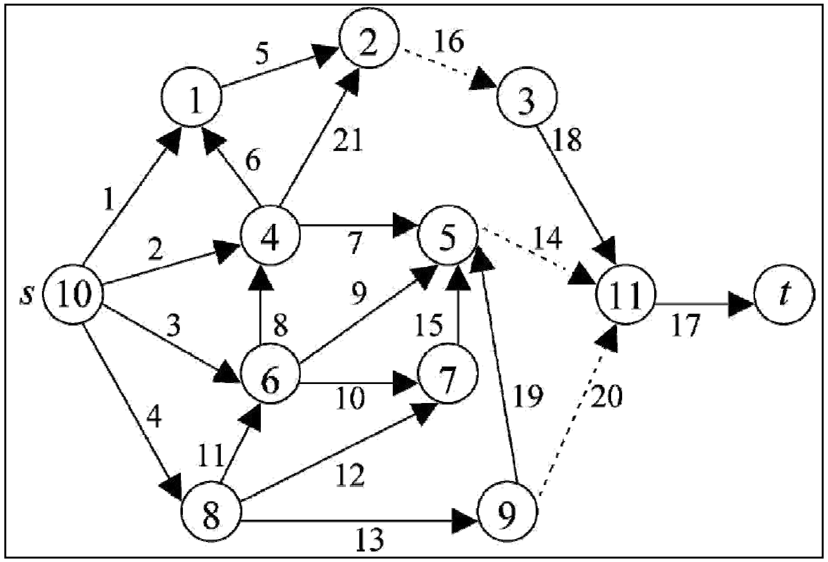


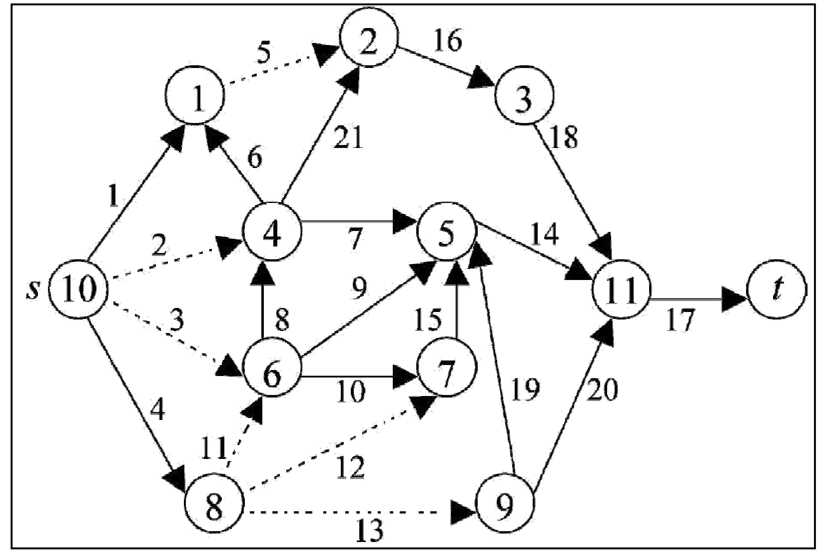
(Ramirez-Marquez and Coit, 2005)

- all 111 MCS's found (10^7 trials, 192 s)

- Most important arcs: 2, 3, 4, 14, 16
  - position with respect to target and source
  - number of downstream arcs (e.g. arcs 1 and 4)
  - failure probability

Influence of failure probability on MCS criticality





MCS 20 $\{14, 16, 20\}$

$frequency = 3 \cdot 10^{-4}$

- More reliable components near target

MCS 7 $\{2, 3, 5, 11, 12, 13\}$

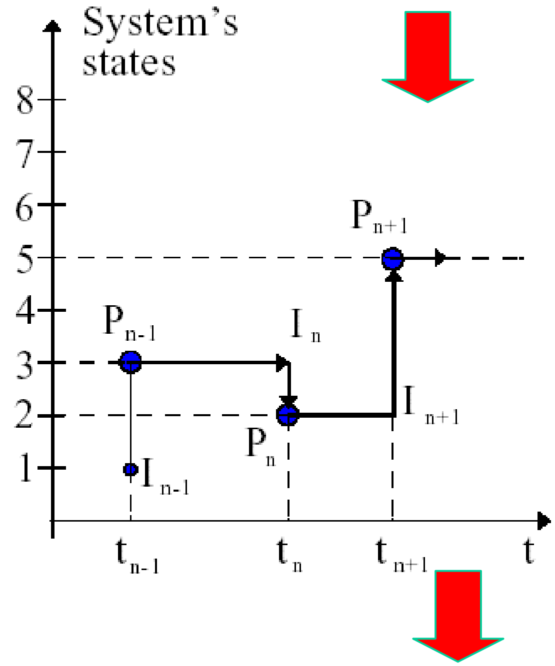
$frequency = 9 \cdot 10^{-4}$

- Less reliable components because of redundancies

BASICS OF MC simulation

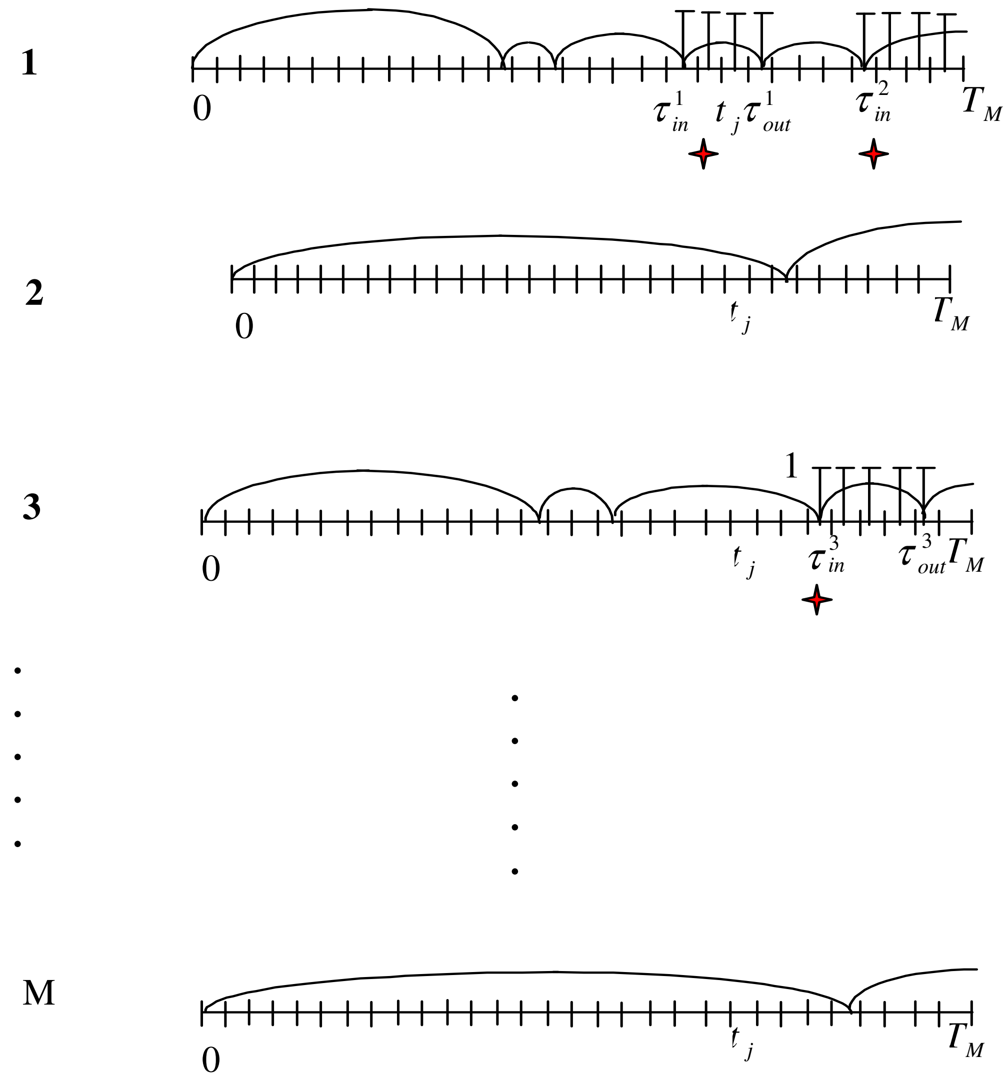
BASICS OF MC Simulation

Procedures for sampling random numbers from given probability distributions



Sampling random walks of arcs failure/repair dynamics

BASICS OF MC Simulation



$$c^A(t_j) = c^A(t_j) + 1$$

$$c^A(t_j) = c^A(t_j)$$

$$c^A(t_j) = c^A(t_j)$$

$$c^A(t_j) = c^A(t_j)$$

$$G^A(t_j) = \frac{c^A(t_j)}{M}$$

Application to network availability

(Zio, Podofillini and Zille, 2005)

Computation of network availability: MC simulation + CA

- Simulate arcs and nodes failure/repair dynamics (MC).

  → sample new configuration of the system after each transition of its elements (nodes and arcs).

- Check system state after each transition (success or failure).

  → check the connectedness between the source and the target nodes (CA).

⇩

Combine Cellular Automata and Monte Carlo Simulation.

Computation of network availability: Method

- The mission time is divided in time channel.
- Simulation of M histories of system life evolution.
- For each history :

  $\Rightarrow$ Sample failure/repair transition times of each network element.

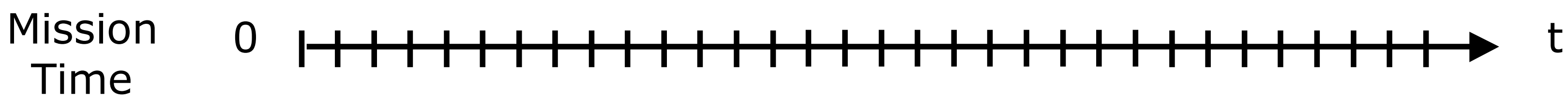
  ⇩

  System configuration in each time channel.

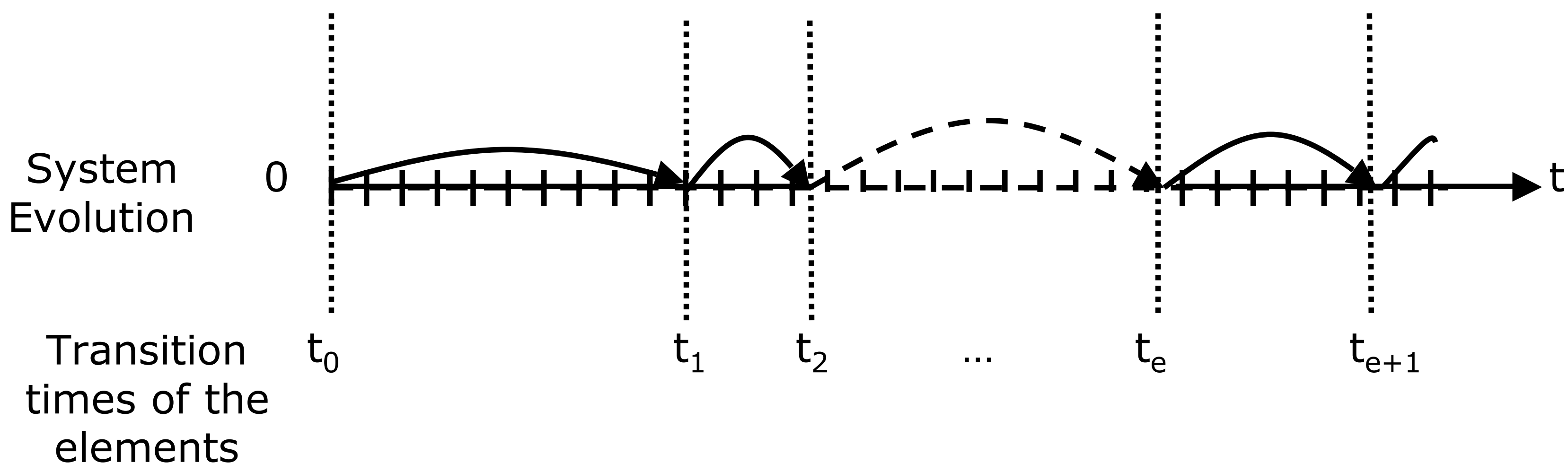
  ⇩

S-T connectedness by CA < successful / failed

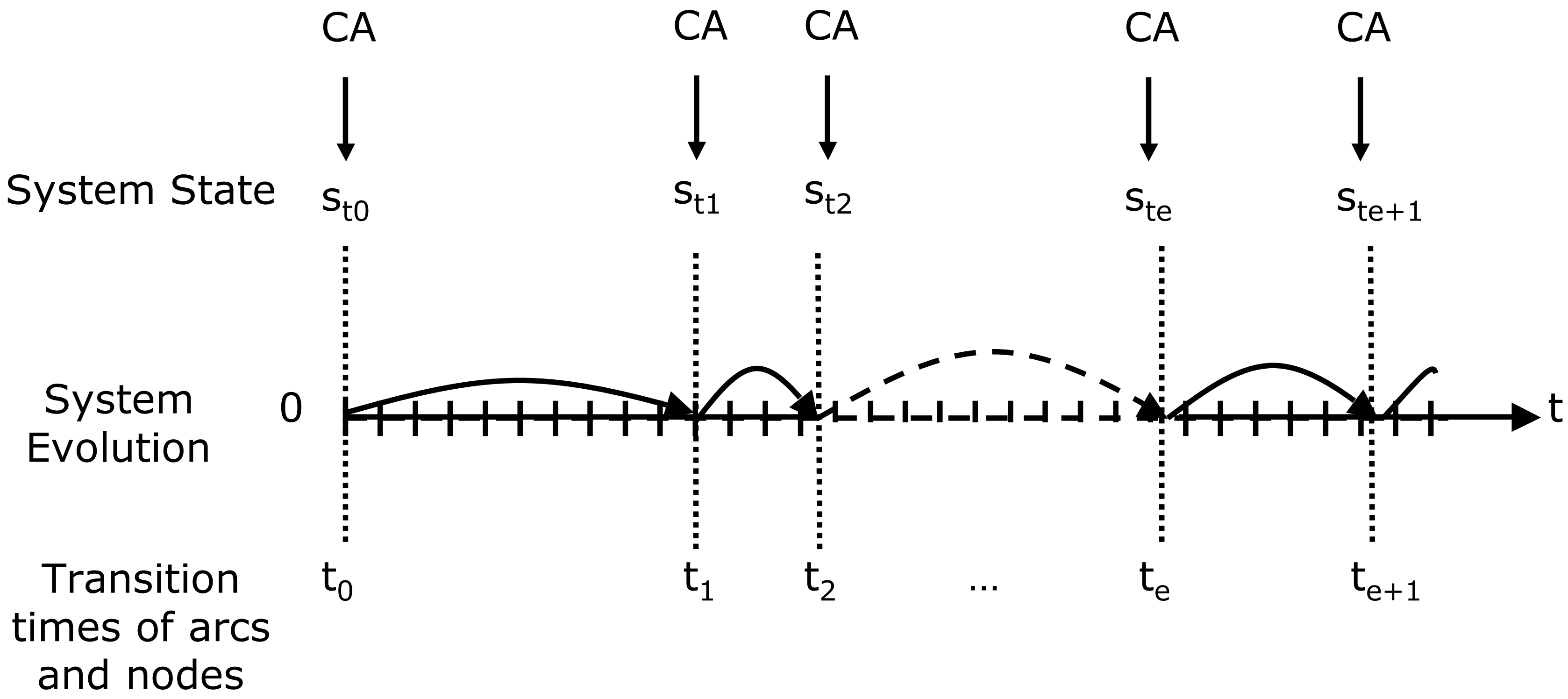
Computation of network availability: Method



Computation of network availability: Method



Computation of network availability: Method



Computation of network availability: Method

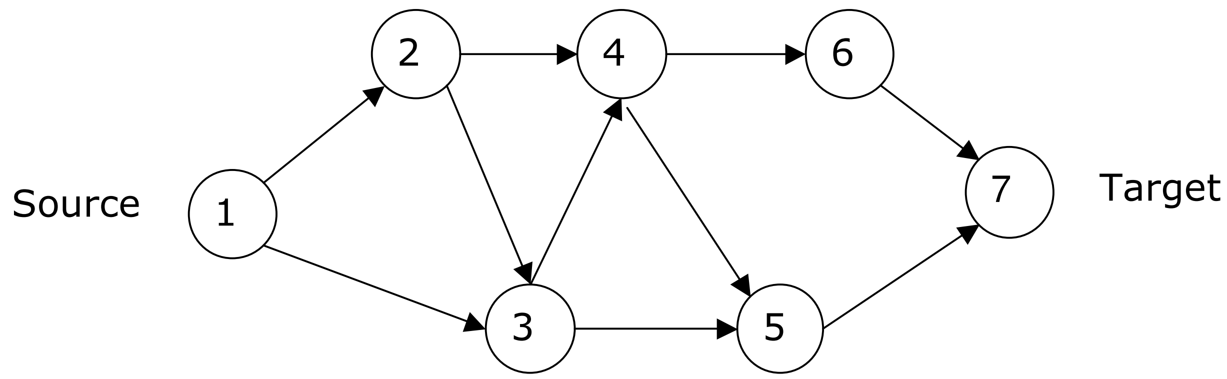
- CA defines the system state after each transition : available or not.

- Collect the portion of time the system is available in the availability counters of the corresponding time channels.

- At the end of the simulation, instantaneous availability of the system :

$$A(t) = \frac{\text{Content of the counter of the corresponding time channel}}{M * dt}$$

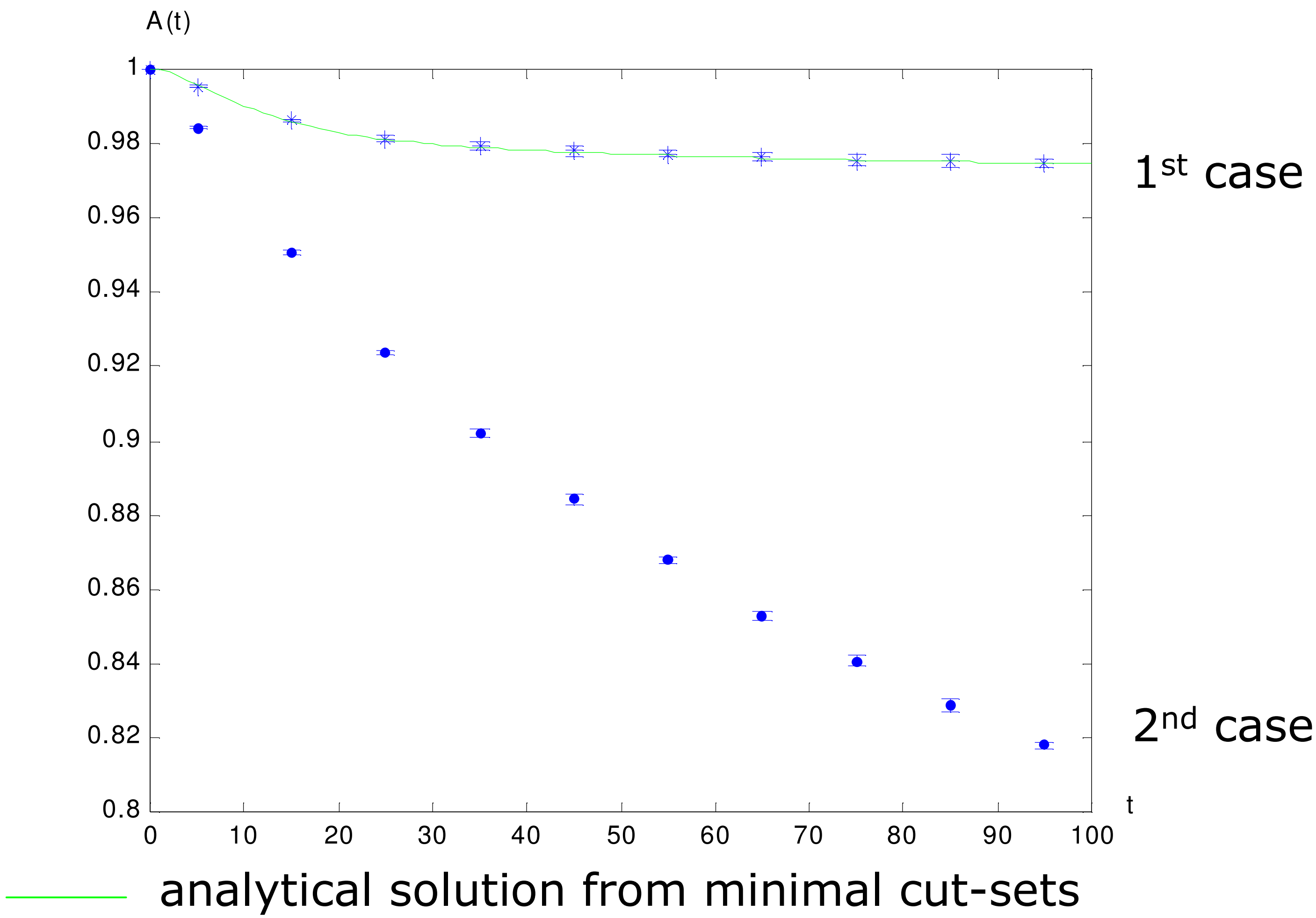
Computation of network availability: Litterature Case Study



1st case : only arcs can fail and be repaired, nodes are assumed perfect.

2nd case : both nodes and arcs can fail and be repaired.

Computation of network availability: Results



CONCLUSIONS

CA+MC:

- ✓ Network connectivity
- ✓ Network reliability
- ✓ Network MCSs e IMs
- ✓ Network Availability

OTHER DEVELOPMENTS

(Zio, Librizzi and Sansavini, 2006)

<u>Binary networks</u> ⟹ <u>Multi-state networks</u>

arc state → 0, 1

arc state → 100% (m_i), 90% (m_i-1), ⋮, 0% (1)

Two-terminal reliability (<u>2TR</u>) ⟹ Multi-state two-terminal reliability at demand level d (<u>M2TR$_d$</u>)

Minimal cut sets (<u>MCS</u>) ⟹ Multi-state minimal cut vectors (<u>MMCV</u>)

Highly reliable networks - Biased Monte Carlo

- High reliability $\Rightarrow$ few failure occurrences $\Rightarrow$ bad statistics
- Biasing $\Rightarrow$ failures favoured $\Rightarrow$ variance reduction

<u>Basic algorithm</u>

i: arc, *j*: state

1. Increase probabilities of arcs in low performing states, $p_{i,j} \rightarrow p^{*}_{i,j}$
2. Sample network configuration from $p^{*}_{i,j}$
3. Compare with network MMCVs (assumed given)
4. If failed configuration $\Rightarrow$ accumulate <u>weight</u>
5. Compute sample estimates of unreliability and variance

$$\prod_{i=1}^{n_{arcs}} \frac{p_{i,j}}{p^{*}_{i,j}}$$

Biased Monte Carlo simulation

Specific biasing

w = mean arc performance in the system

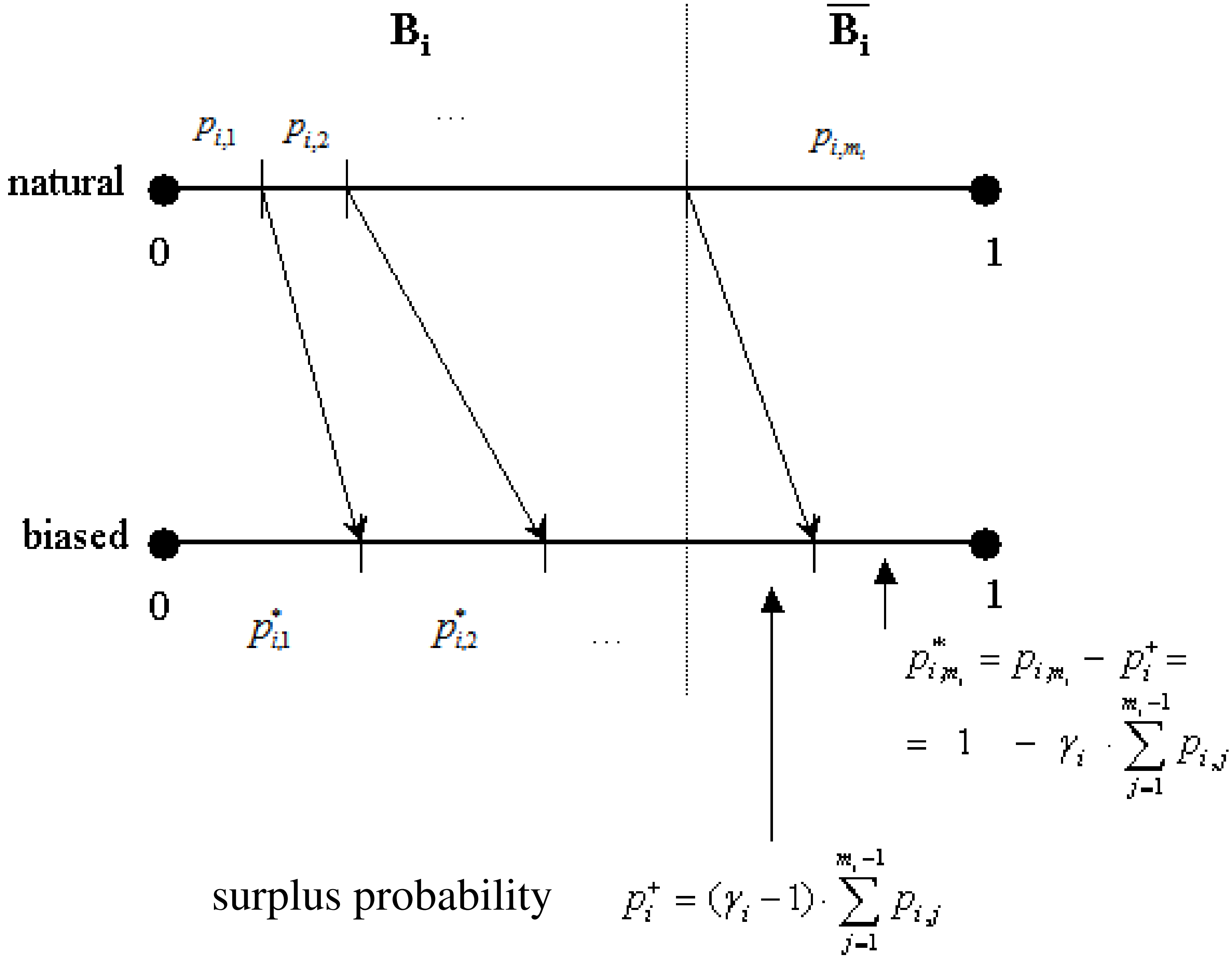
- Set threshold for biasing: $w_{th} = k \cdot w$

- Bias arc i with $w_i > w_{th}$ so that $w_i \le w_{th}$

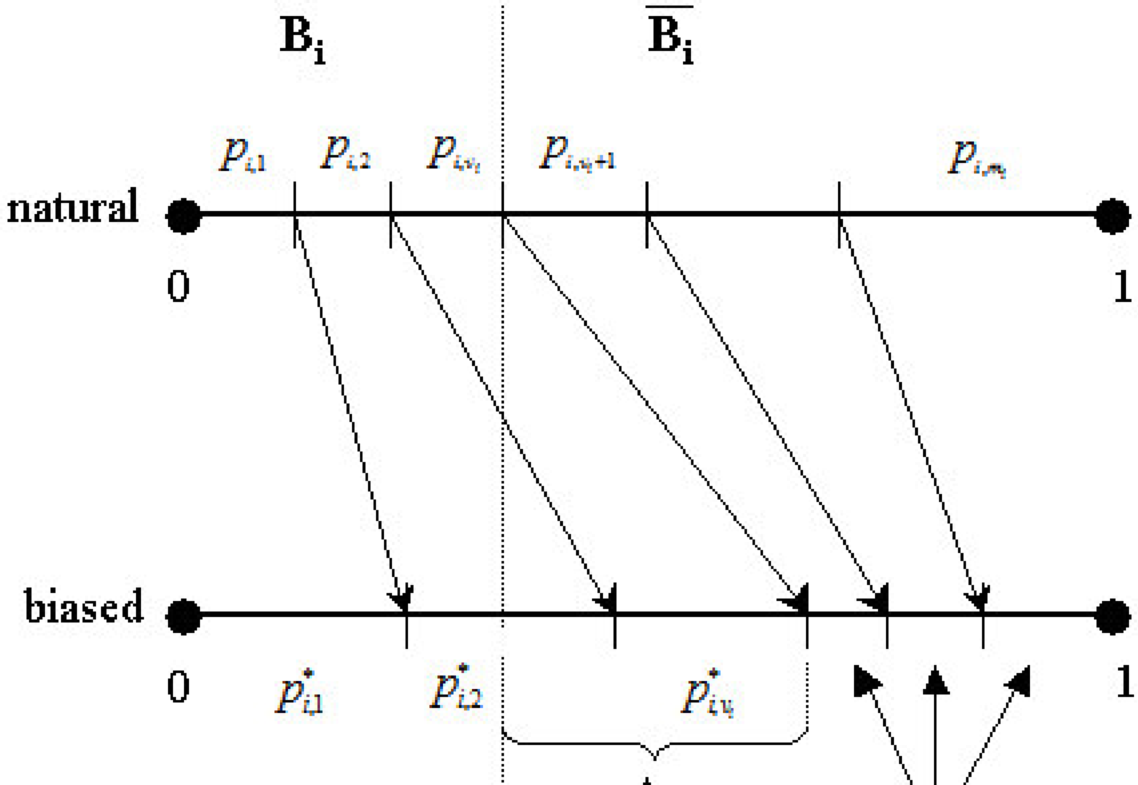
Method 1

Method 2

Biasing method 1 – all states below the nominal one



Biasing method 2 – all states below the minimal in MMCV



$$p_i^+ = (\gamma_i - 1)\cdot \sum_{j\in B_i} p_{i,j}$$

$$p^*_{i,j} = p_{i,j} - \frac{p_{i,j}}{\sum_{j'\in \overline{B_i}} p_{i,j'}}\cdot p_i^+ = p_{i,j}\cdot\left[\frac{1-\gamma_i\cdot\sum_{j'\in B_i} p_{i,j'}}{\sum_{j'\in \overline{B_i}} p_{i,j'}}\right]$$

only failed states are forced

Conclusions

- Need for efficient computational techniques for assessing highly reliable, multi-state networks $\Rightarrow$ biased MC

Current developments

• Extension CA+MC for application to Security

Zio & Rocco, CNIP' 06, 2006

• Development of new reliability indicators from the evaluation of the network topological properties (Complexity Science)

Zio, 2006

Future developments

MC biasing based on MMCVs, which are difficult to identify

⇓

Flow algorithms, CA (?)

Cellular Automata (CA)

- **WHAT?**
  - ✓ Mathematical models of complex dynamical systems
  - ✓ Large number of identical processing elements with local interactions
  - ✓ Parallel computation
- **WHY?**
  - ✓ Development of computers and computation (von Neumann, 1948)
  - ✓ Models of the dynamics of many real complex systems: e.g. fluids, molecular systems, economical systems, ecological systems

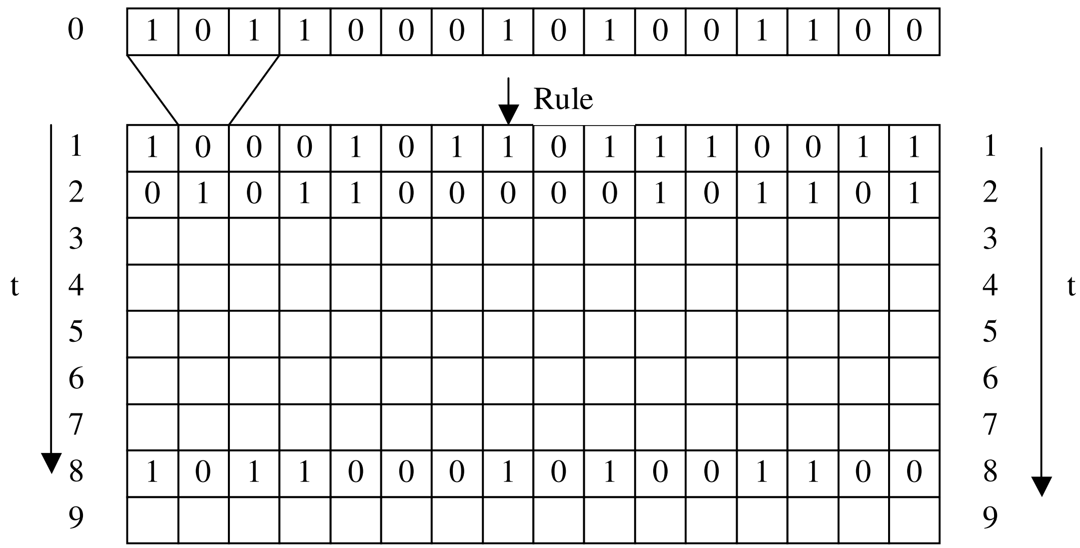
BASICS OF CA

BASICS OF CA

1. Spatially- and temporally-discrete
2. Local interaction
3. Parallel evolution

BASICS OF CA
1. Spatially- and temporally-discrete

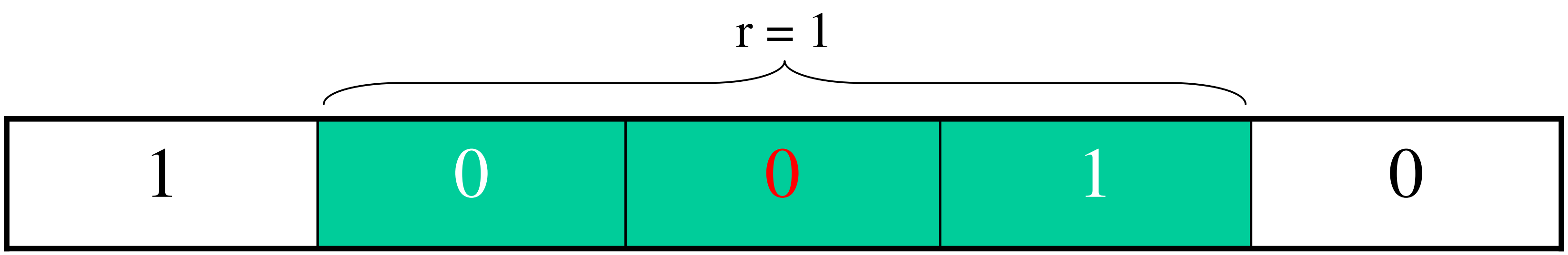
- ✓ $\mathcal{L}$ = discrete lattice of cells (state space for CA dynamics)



- ✓ Each cell of $\mathcal{L}$ is a *finite automaton* which assumes values in S≡{0,1,2,…,k-1}
- ✓ $s_i(t)$= state of the cell *i (1D)* at the discrete time *t*
- ✓ $\mathcal{L}$ is homogeneous: all cells bear the same properties

BASICS OF CA
2. Local interaction

- ✓ N_i = predefined local neighbourhood of cell i



- ✓ Cell i interacts only with the n cells in N_i
- ✓ Transition rule

$$\phi : \overbrace{S \, X \, S \, X \ldots X \, S}^{n} \rightarrow S$$

$$s_i(t+1) = \phi[s_r(t), \quad r \in N_i]$$

BASICS OF CA
3. Parallel evolution

- ✓ One evolution step of the CA is achieved after the simultaneous application of the rule ϕ to each cell in $\mathcal{L}$

1D-CA Example: Addition modulo 2

- ✓ S≡{0,1}; k=2
- ✓ r=1
- ✓ 1st order in time

$$s_i(t+1) = \mathrm{mod}_2\left[s_{i-1}(t) + s_i(t) + s_{i+1}(t)\right] \equiv \oplus_2\left\{s_{i-1}(t), s_i(t), s_{i+1}(t)\right\}$$

$i-1(t), i(t), i+1(t)$	1,1,1	1,1,0	1,0,1	1,0,0	0,1,1	0,1,0	0,0,1	0,0,0
$i(t+1)$	1	0	0	1	0	1	1	0

1D-CA Example: Addition modulo 2

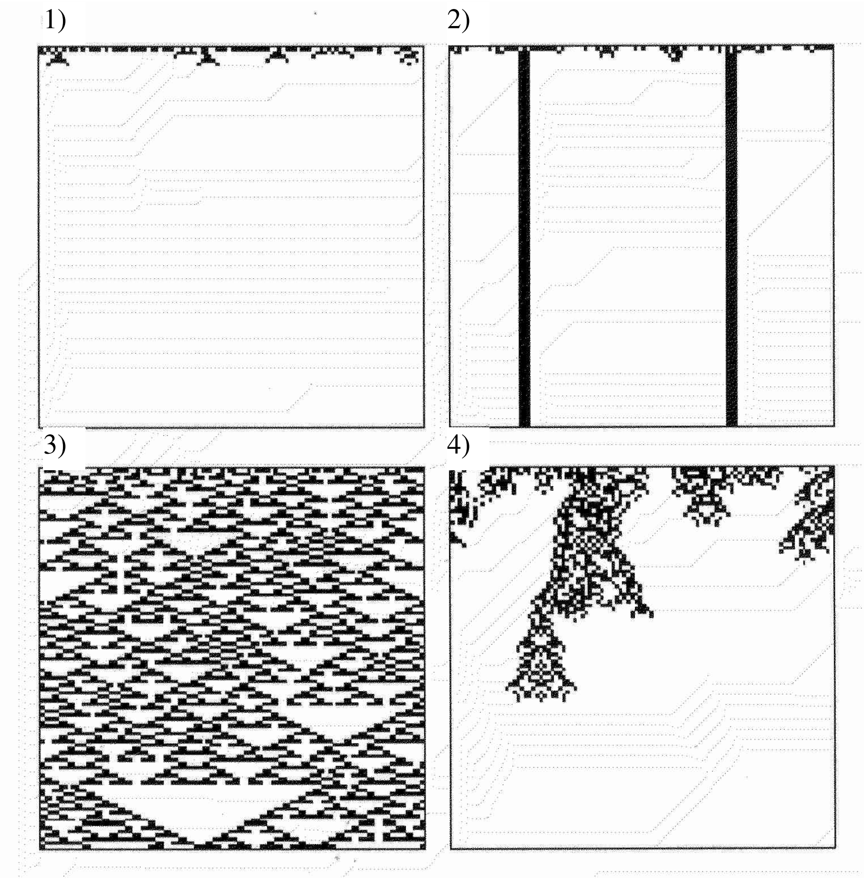
t																
0	1	0	1	1	0	0	0	1	0	1	0	0	1	1	0	0
1	1	0	0	0	1	0	1	1	0	1	1	1	0	0	1	1
2	0	1	0	1	1	0	0	0	0	0	1	0	1	1	0	1
3																
4																
5																
6																
7																
8	1	0	1	1	0	0	0	1	0	1	0	0	1	1	0	0
9																

Rule

t

CA Behavioural classes

1. fixed points
2. inhomogeneous configuration or cycles
3. chaotic, aperiodic patterns
4. complex, localized, propagating structures



CA vs DE

- ✓CA can be considered an alternative to DE
- ✓Discretization for numerical solution of DE ~ local discrete dynamical system of CA
- ✓DE can lead to analytic solutions in simple cases
- ✓CA are more convenient for simulation
- ✓HOWEVER: setting up a CA corresponding to a DE is a difficult problem ⇨phenomenology

BASICS OF MC sampling

BASICS OF MC Sampling

Procedures for sampling random numbers from given probability distributions

↓

"Coin Toss":

p_{ji} = probability of arc *ji* "success"

q_{ji} = probability of arc *ji* "failure"

↓

Sampling realizations of arcs failure/success configurations