

CONNETTIVITA' E AFFIDABILITA' DI SISTEMI COMPLESSI

Andrea Bobbio Roberta Terruggia

Dipartimento di Informatica

Università del Piemonte Orientale,

“A. Avogadro”

15100 Alessandria (Italy)



Studiare una rete

La struttura di un sistema può essere rappresentata tramite un grafo $G=(V,E)$ dove V è l'insieme dei nodi ed E l'insieme degli archi.

Vogliamo studiare:

- **Connettività**: la funzione booleana che rappresenta come a partire da un nodo sorgente è possibile raggiungere un nodo destinazione.
 - Minpath
 - Mincut
- **Affidabilità**: probabilità che due nodi siano connessi

Archi :

- Diretti;
- Non diretti

Possono guastarsi:

- Solamente gli archi;
- Solamente i nodi;
- Sia gli archi che i nodi.

Path e Cut

Dato un grafo $G=(V,E)$ un **path** H è definito come un insieme di componenti nodi e/o archi che se funzionanti permettono di connettere il nodo sorgente con il nodo destinazione.

Un path è minimo (**minpath**) se non esiste al suo interno un sottoinsieme che sia anch'esso un path.

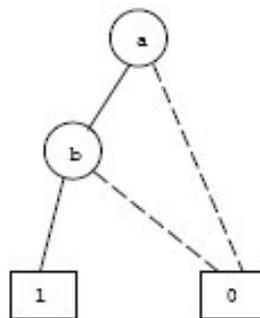
Dato un grafo $G=(V,E)$ un **cut** K è definito come un insieme di componenti nodi e/o archi che se guaste disconnettono il nodo sorgente dal nodo destinazione.

Un cut è minimo (**mincut**) se non esiste al suo interno un sottoinsieme che sia anch'esso un cut.

Binary Decision Diagrams

Un BDD è costituito da un albero binario composto da due tipi di vertici:

- **Terminali**: costituiti dal nodo 1 o dal nodo 0 e rappresentano i valori costanti true e false (sistema funzionante o guasto)
- **Non terminali**: rappresentano le componenti ed hanno due archi uscenti etichettati con i valori di 0 (linea tratteggiata) ed 1 (linea continua) a seconda che si consideri la componente funzionante oppure guasta.

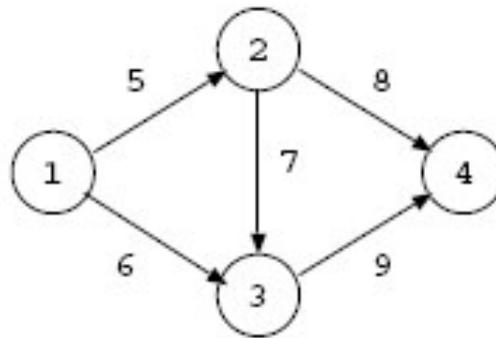


Costruzione del BDD a partire dal grafo

```
bdd_gen(start_node){
  T_bdd=NULL
  inserisco start_node in this_path
  for(edge_i=ogni arco uscente da start_node){
    next_node=nodo a cui start_node risulta collegato tramite
    edge_i
    if(next_node=destinazione)
      subpath=edge_i
    else if(next_node già presente in this_path)
      continua
    else
      sub_path=bdd_gen(next_node)AND edge_i_bdd
    T_bdd=T_bdd OR subpath_bdd
  }
  elimino start_node da this_path
  return T_bdd
}
```

Tratto dalla tesi di dottorato di Xinyu Zang

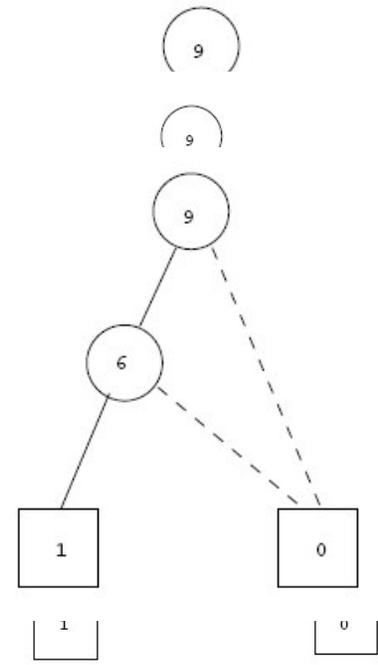
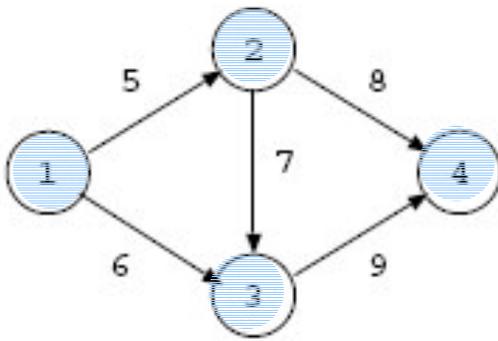
Esempio: Rete Bridge



Ordinamento archi:

$9 \prec 7 \prec 8 \prec 5 \prec 6$

Costruire il BDD

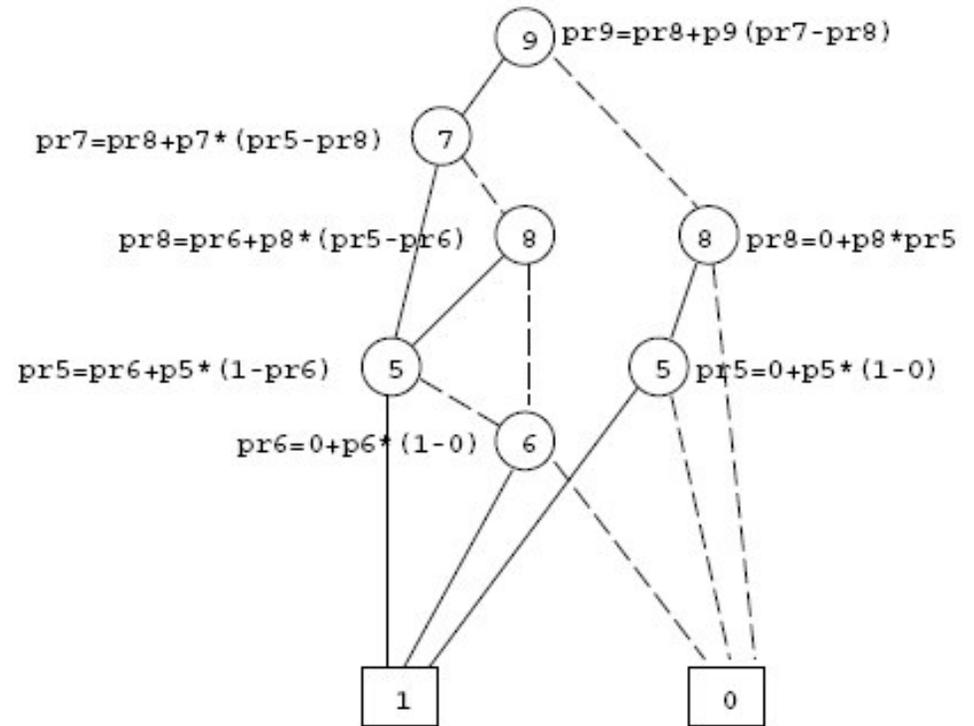


Calcolo Affidabilità

```

Prob(F){
  if(F==0)
    return 0
  else if(F==1)
    return 1
  else if(computed-table has entry {F,P_F})
    return P_F
  else { /*F=ite(x,F1,F2)
    P_F1=Prob(F1)
    P_F2=Prob(F2)
    P_F=P_F2+P(x)*(P_F1-P_F2)
  }
  insert_comute_table({F,P_F})
  return P_F
}

```



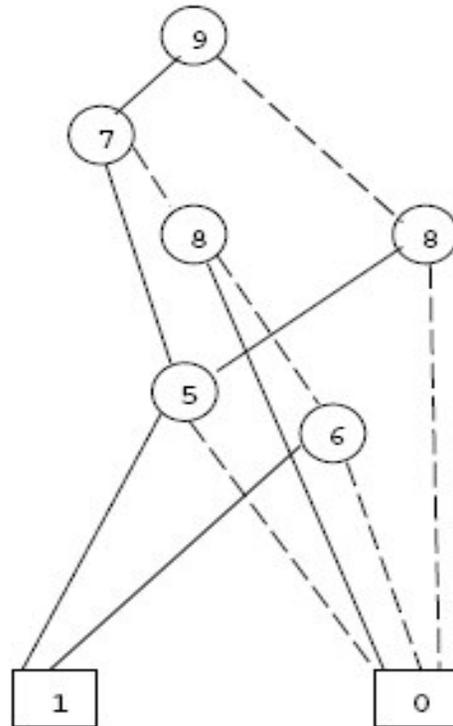
$$P(F) = P(F_{x=0}) + p_x(P(F_{x=1}) - P(F_{x=0}))$$

Calcolo Minpath e Mincut

```
minbdd(F,flag){
  if(F==1) or (F==0)
    return F
  else if(computed-table has entry{(flag,F),R})
    return R
  else{/* F=ite(x,F1,F2)*/
    if(flag==1){ /* minpath */
      K=minbdd(F1,flag)
      T=without(K,F2,1)
      E=minbdd(F2,flag)
    }else { /* mincut */
      T=minbdd(F1,flag)
      K=minbdd(F",flag)
      E=without(K,F1,0)
    }
  }
  if(T==E)
    return T
  R=find_or_add_unique_table(x,T,E)
  insert_computed_table({(flag,F),R})
  return R
}
```

Antoine Rauzy e Xinyu Zang

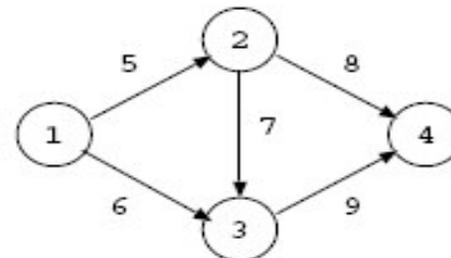
BDD minimo per il calcolo dei Minpath



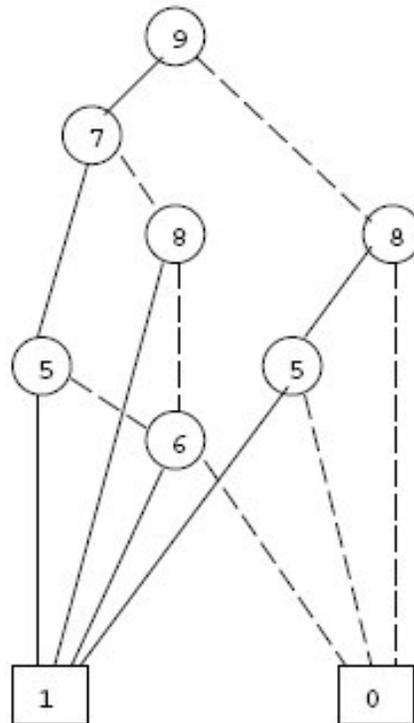
Il path {9,8,5} non è minimo... contiene il minpath {8,5}

Il path {9,7,6} non è minimo... contiene il minpath {9,6}

Minpath: {9,7,5}, {5,8}, {9,6}

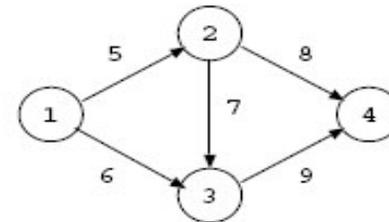


BDD minimo per il calcolo dei Mincut

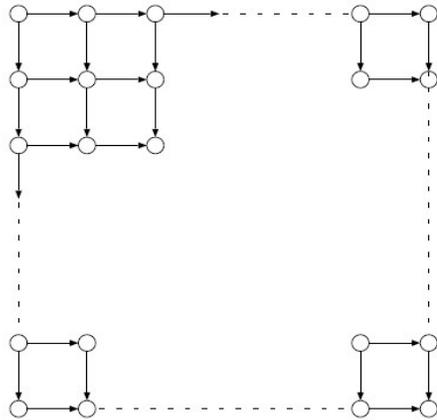


Il cut $\{7,5,6\}$ non è minimo... contiene il mincut $\{5,6\}$

Mincut: $\{9,8\}$, $\{9,5\}$, $\{7,8,6\}$, $\{5,6\}$



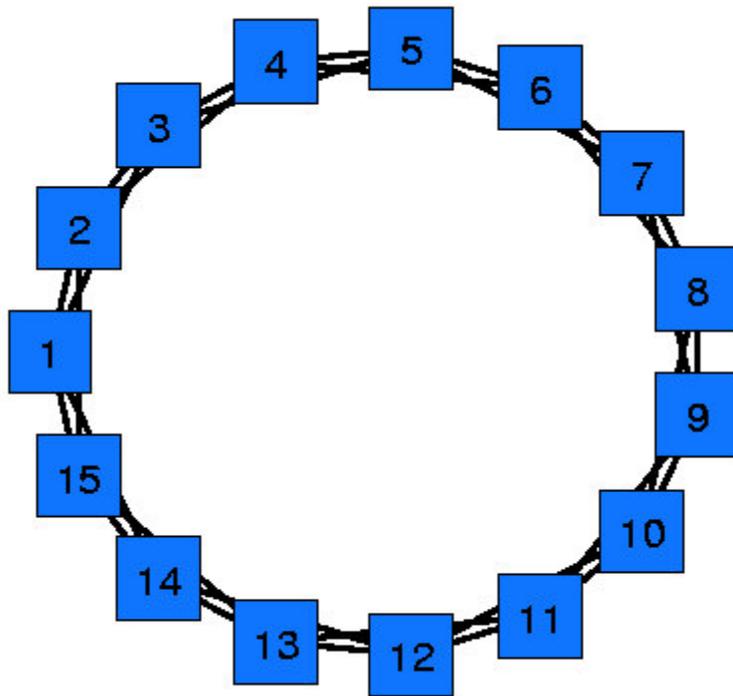
Analisi di una rete regolare: il Lattice



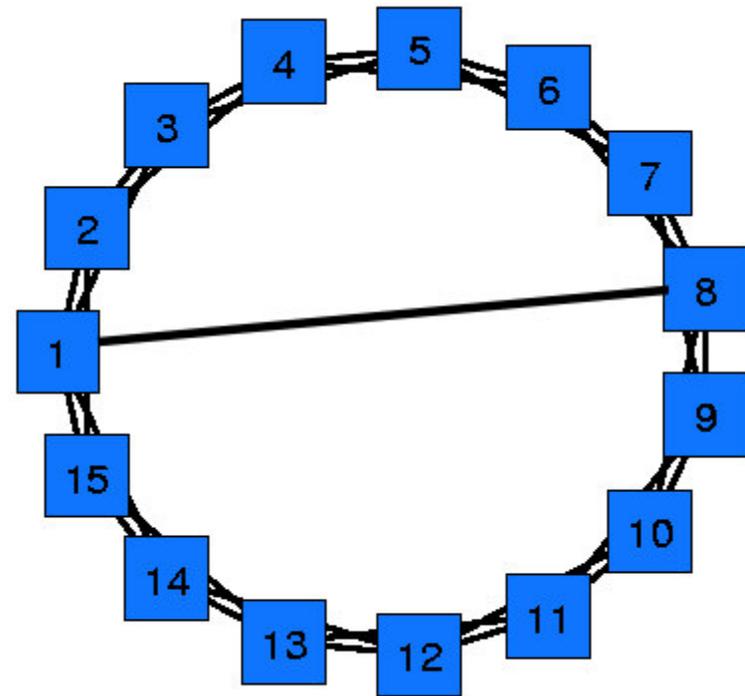
Lattice	# nodi Lattice	# archi Lattice	# nodi BDD	# minpath
2 X 2	4	4	6	2
4 X 4	16	24	94	20
6 X 6	36	60	1034	252
8 X 8	64	112	8384	3432
10 X 10	100	180	56338	N.A.
12 X 12	144	264	342038	N.A.

Il BDD cresce in modo esponenziale rispetto ad n !!!

Reti regolari e reti small world dirette

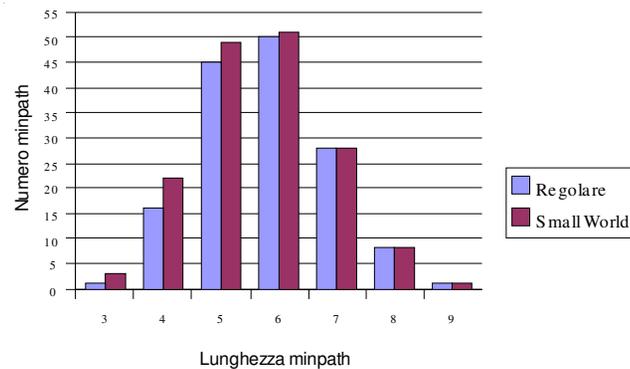


Rete regolare

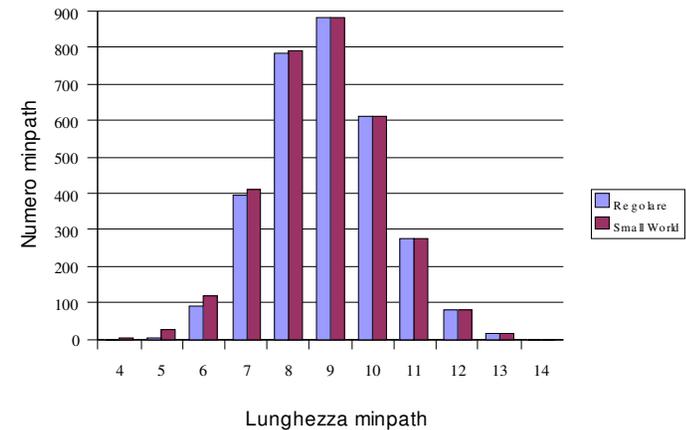


Rete Small World

Reti regolari e reti small world dirette



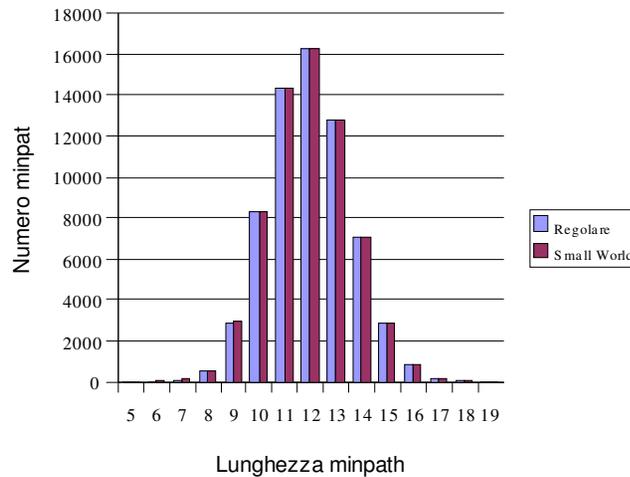
10 nodi



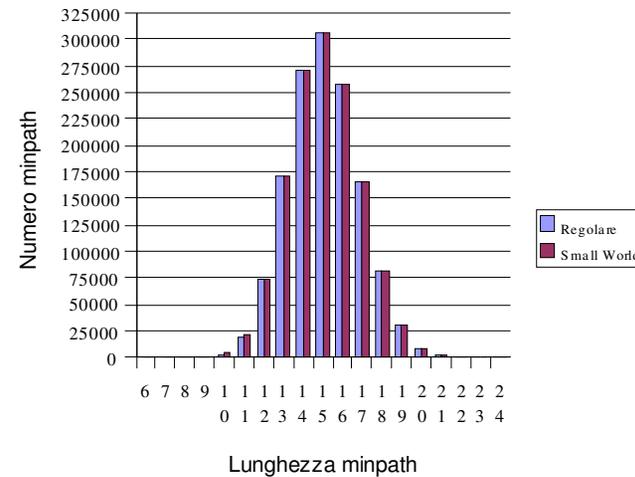
15 nodi

	R.regolare	R. small world	R.regolare	R. small world
n. nodi BDD	81	88	141	172
n. minpath	149	162	3136	3217
n. mincut	112	122	1286	1462
Prob	0.997936	0.998862	0.997931	0.998859

Reti regolari e reti small world dirette



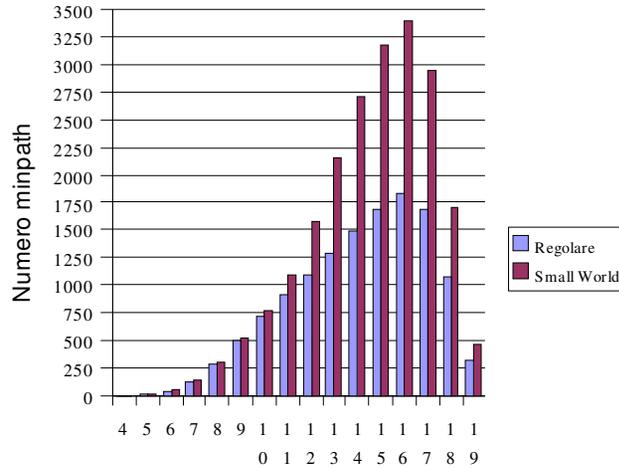
20 nodi



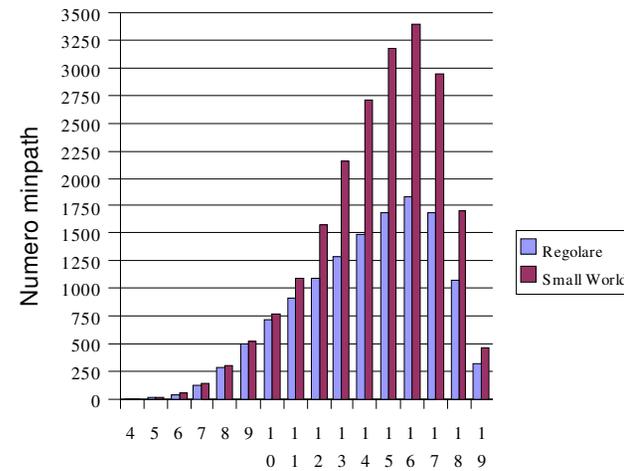
25 nodi

	R.regolare	R. small world	R.regolare	R. small world
n. nodi BDD	201	268	261	352
n. minpath	66012	66286	1389537	1391242
n. mincut	14321	17105	158896	192420
Prob	0.997925	0.998857	0.997920	0.998853

Reti regolari e reti small world non dirette



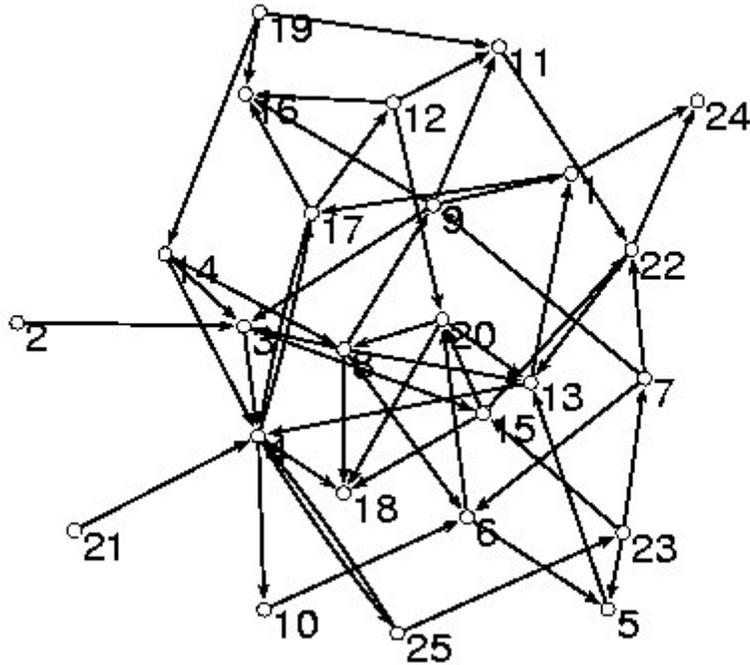
Lunghezza minpath
15 nodi



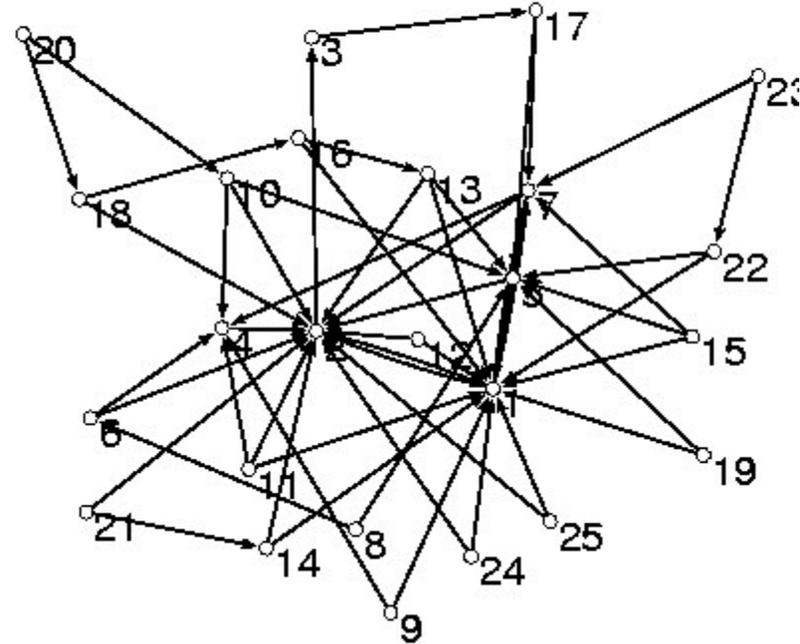
Lunghezza minpath
20 nodi

	R.regolare	R. small world	R.regolare	R. small world
n. nodi BDD	960	1004	1472	1596
n. minpath	1834	2462	13002	20984
n. mincut	508	648	1756	2402
Prob	0.999745	0.998859	0.999701	0.999830

Reti Scale Free dirette

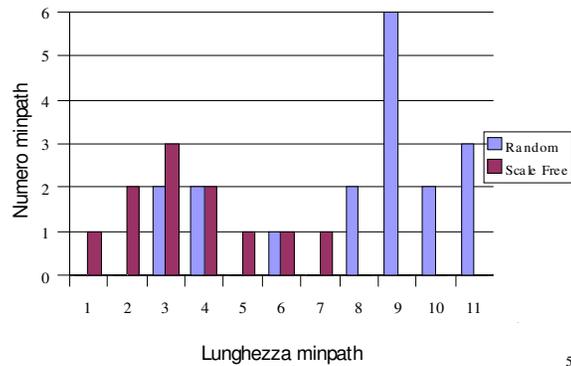


Rete Random 54 nodi

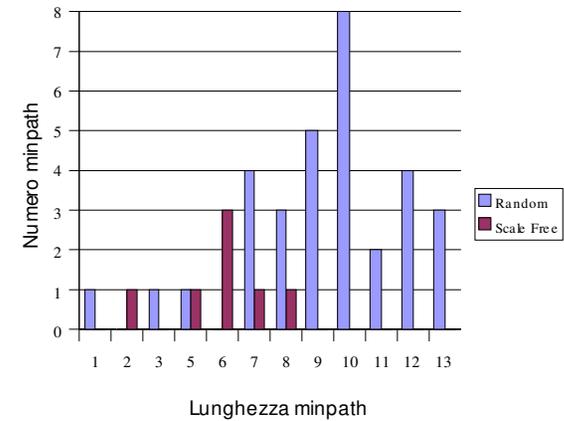


Rete Scale Free 54 nodi

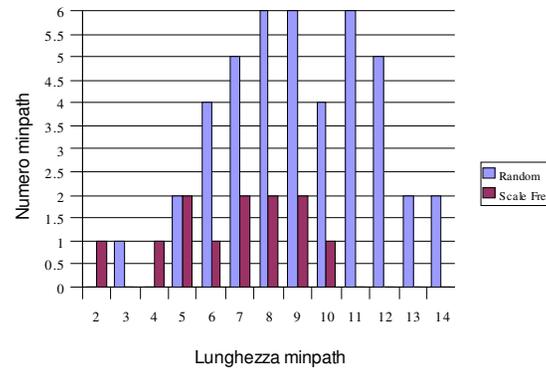
Reti Scale Free dirette



S=15 D=1



S=23 D=5



S=20 D=5

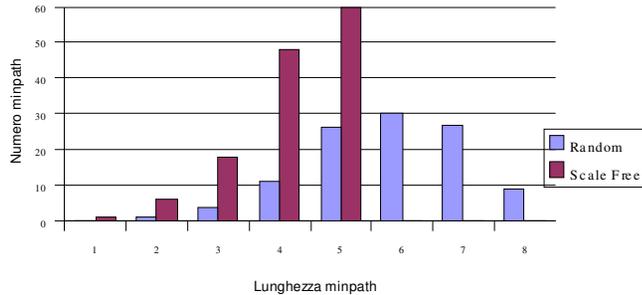
S	D	Nodi BDD	n. minpath	n. mincut	Prob.
15	1	26	11	11	0.998844
23	5	28	7	17	0.940135
11	18	-	-	-	-
7	20	-	-	-	-
20	5	48	12	47	0.974142
18	2	33	15	15	0.988985

Rete Scale free

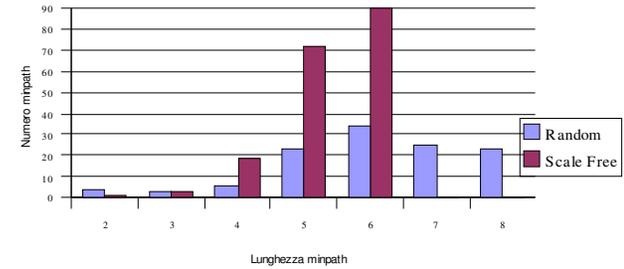
S	D	Nodi BDD	n. minpath	n. mincut	Prob.
15	1	313	18	231	0.962343
23	5	307	32	238	0.988221
11	18	112	25	136	0.780340
7	20	304	19	331	0.992056
20	5	278	43	105	0.954186
18	2	-	-	-	-

Rete Random

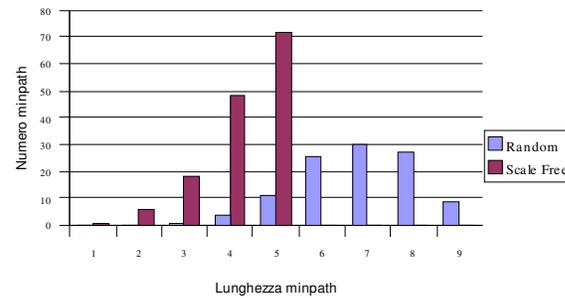
Reti Scale Free non dirette



S=7 D=4



S=6 D=2



S=3 D=7

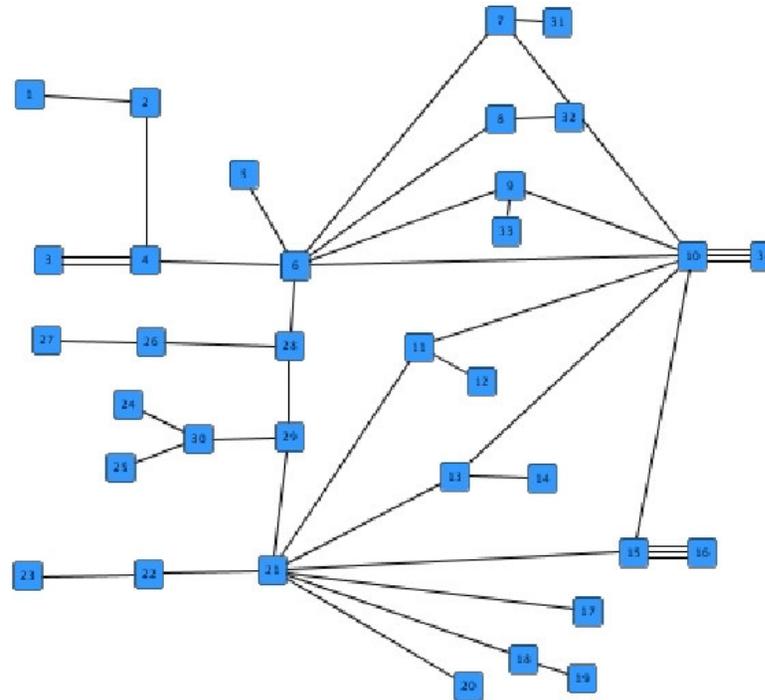
S	D	Nodi BDD	n. minpath	n. mincut	Prob.
1	10	-	-	-	-
7	4	45	133	10	0.999899
5	8	-	-	-	-
6	2	77	184	12	0.899908
3	7	64	145	10	0.999899

Rete Scale free

S	D	Nodi BDD	n. minpath	n. mincut	Prob.
1	10	235	103	58	0.997890
7	4	340	108	58	0.988988
5	8	192	56	54	0.999899
6	2	206	118	56	0.998711
3	7	358	108	59	0.890089

Rete Random

Schema rete UPO



REFERENCES

- [ALBE02a] - R. Albert and A.L. Barabasi. Statistical mechanics of complex networks. *Review Modern Physics*, 74:47--97, 2002.
- [BARA02a] - A.L. Barabasi. Linked: the new science of networks. 2002.
- [BRYA86] - R.E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35:677--691, 1986.
- [HARD05] - G. Hardy and C. Lucet and N. Limnios, Computing all-terminal reliability of stochastic networks by Binary Decision Diagrams, *Proceedings Applied Stochastic Modeling and Data Analysis, ASMDA2005*, 2005
- [KAUF77] - A. Kaufmann, D. Grouchko, and R. Cruon. Mathematical Models for the Study of the Reliability of Systems. Academic Press, 1977.
- [NEWM03a] - M.E. Newman. The structure and function of complex networks. *SIAM Review*, 45:167--256, 2003.
- [RAUZ93] - A. Rauzy, New algorithms for fault tree analysis, *Reliability Engineering and System Safety*, 40, 203-211, 1993
- [TRIV01a] - K. Trivedi. Probability & Statistics with Reliability, Queueing & Computer Science applications, Wiley, II Edition, 2001.
- [ZANG00] - X. Zang, H. Sun, K. Trivedi, A BDD-based algorithm for reliability graph analysis, Department of Electrical Engineering, Duke University, 2000