Average Consensus con Perdita di Pacchetto

Moscatelli Luca
Sciortino Alessandro

Università degli Studi di Padova

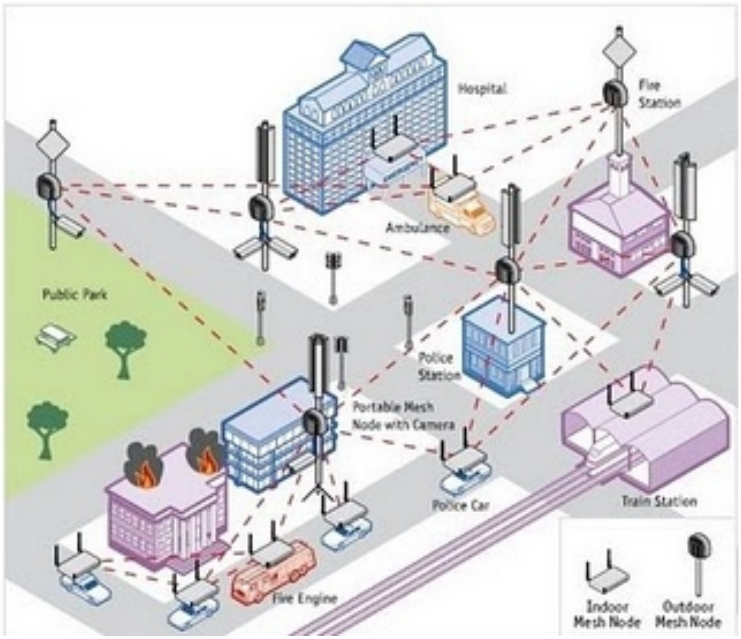
27 marzo 2012

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Indice

Formulazione del Problema

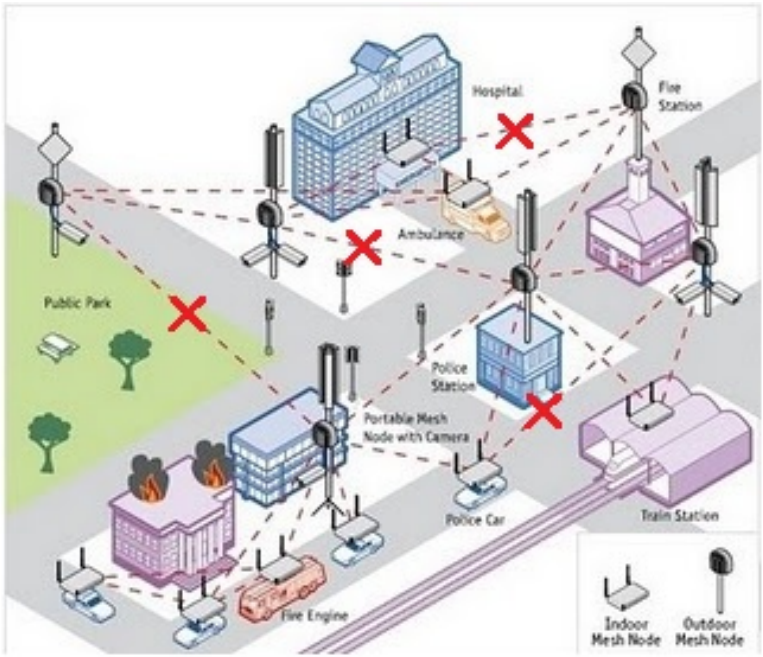
- Si supponga che vari agenti di una stessa struttura vogliano comunicare fra loro per scambiarsi informazioni



- Il loro obiettivo sia il raggiungimento di un valore di stima comune di una certa misura

Formulazione del Problema

- Gli algoritmi che permettono di raggiungere la media degli stati sono chiamati algoritmi di *average consensus*
- Nel caso, ad esempio, di una rete di comunicazione wireless, è possibile che alcuni pacchetti di dati vadano persi



Obiettivi

- L'obiettivo di questo progetto è introdurre la perdita di pacchetto in alcuni algoritmi di average consensus, e valutarne le prestazioni in termini di errore e velocità di convergenza
- I confronti e le analisi saranno basati esclusivamente sui risultati delle simulazioni, non è stata sviluppata alcuna teoria scientifica

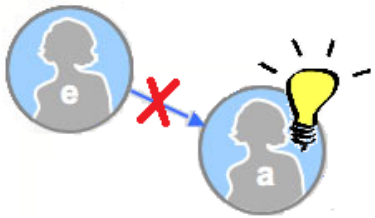
Approccio Proposto

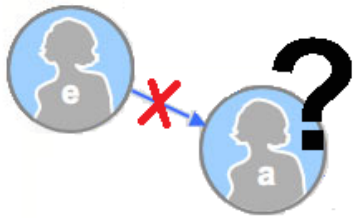
- Sono stati presi in esame algoritmi per reti sincrone ed asincrone
- Gli algoritmi per reti asincrone si suddividono a seconda della tipologia di comunicazione: si vedranno due algoritmi broadcast e un algoritmo gossip
- Quasi tutti gli algoritmi presi in considerazione prevedono l'introduzione di variabili aggiunte denominate *stati aumentati*
- L'introduzione della perdita di pacchetto è stata realizzata tramite delle variabili aleatorie binomiali che rappresentano la probabilità che una certa comunicazione vada a buon fine

Approccio Proposto

Per gli algoritmi broadcast sono state distinte due possibili varianti: il caso consapevole ed il caso inconsapevole

- Nel caso consapevole l'agente che avrebbe dovuto ricevere l'informazione è a conoscenza della tentata trasmissione
- Nel caso inconsapevole l'agente che avrebbe dovuto ricevere l'informazione non è a conoscenza della tentata trasmissione





Algoritmi Sincroni

ASP è l'unico algoritmo a stato non aumentato:

- prevede già la perdita di pacchetto
- metodo della compensazione parziale
- metodo della compensazione bilanciata

Algoritmo AS:

- algoritmo a stato aumentato

ASP

- L'unico a prevedere una compensazione della perdita dei pacchetti
-

$$\mathbb{P}[L_{ij}(t) = 1] = p \qquad \mathbb{P}[L_{ij}(t) = 0] = 1 - p$$

Metodo della compensazione parziale

$$x_i(t+1) = \left(P_{ii} + \sum_{j \neq i} (1 - L_{ij}(t)) P_{ij} \right) x_i(t) + \sum_{j \neq i} L_{ij}(t) P_{ij} x_j(t)$$

Metodo della compensazione bilanciata

$$x_i(t+1) = \frac{1}{P_{ii} + \sum_{j \neq i} L_{ij}(t) P_{ij}} \left(P_{ii} x_i(t) + \sum_{j \neq i} L_{ij}(t) P_{ij} x_j(t) \right)$$

AS

- Algoritmo originale:

$$x_i(t+1) = x_i(t) + \sum_{j \in \mathcal{N}_i^+} a_{ij}(x_j(t) - x_i(t)) + \epsilon s_i(t)$$

$$s_i(t+1) = \Big((1 - \sum_{h \in \mathcal{N}_i^-} b_{ih}) s_i(t) + \sum_{j \in \mathcal{N}_i^+} b_{ji} s_j(t)\Big) - \Big(x_i(t+1) - x_i(t)\Big)$$

- Pesi archi:

$$a_{ij} = \frac{1}{(\mathrm{card}(\mathcal{N}_i^+)+1)} \qquad b_{ih} = \frac{1}{(\mathrm{card}(\mathcal{N}_i^-)+1)}$$

- ϵ=0.2

AS con perdita di pacchetto

- Introduzione perdita pacchetto

$$x_i(t+1) = x_i(t) + \sum_{j \in \mathcal{N}_i^+} a_{ij} p_{ji}(t)(x_j(t) - x_i(t)) + \epsilon s_i(t)$$

$$s_i(t+1) = \Big((1 - \sum_{h \in \mathcal{N}_i^-} b_{ih}))s_i(t) + \sum_{j \in \mathcal{N}_i^+} b_{ji} p_{ji}(t) s_j(t)\Big) - \Big(x_i(t+1) - x_i(t)\Big)$$

Algoritmi Asincroni

Sono stati presi in considerazione tre algoritmi asincroni (tutti a stato aumentato) che si differenzieno per il tipo di comunicazione su cui sono basati:

- AB è la generalizzazione di AS per reti asincrone, ed è pensato per comunicazione broadcast
- AGB è un algoritmo in cui ogni coppia di nodi in cui un agente sia sufficientemente distante dall'altro può avviare una comunicazione di tipo broadcast, senza che la trasmissione di uno dei due nodi interferisca con quella dell'altro
- AG è un algoritmo basato su comunicazione gossip

AB

A cambiare rispetto all'algoritmo originale è solo l'aggiornamento dei nodi riceventi, mentre la dinamica dei nodi che trasmettono e dei nodi inattivi rimane la stessa.
Sono state prese in considerazioni due possibili varianti:

- Caso consapevole: algoritmo originale:

$$x_j(t+1) = \frac{x_i(t) + x_j(t)}{2} + \epsilon s_j(t) \tag{1}$$

algoritmo con perdita di pacchetto:

$$x_j(t+1) = \frac{x_j(t)}{2} + \epsilon s_j(t) \tag{2}$$

AB

- Caso inconsapevole: l'aggiornamento dei nodi riceventi avviene allo stesso modo dei nodi isolati:

$$x_j(t+1) = x_j(t) + \epsilon s_j(t) \tag{3}$$

In entrambi i casi l'aggiornamento dello stato aumentato avviene come nei nodi isolati dell'algoritmo originale:

$$s_j(t+1) = s_j(t) - \Big(x_j(t+1) - x_j(t)\Big) \tag{4}$$

Se non si perde informazione anche l'aggiornamento dei nodi riceventi avviene come nell'algoritmo originale.

AGB

Anche in questo caso a cambiare rispetto all'algoritmo originale è solo l'aggiornamento dei nodi riceventi.
Come in AB sono stati distinti il caso consapevole ed il caso inconsapevole:

- Caso consapevole: algoritmo originale:

$$\begin{cases} x_j(t+1) = \frac{x_j(t)+x_i(t)}{2} + 0.5 s_j(t) + \frac{s_i(t)}{2\delta_{out,i}(t)} \\ s_j(t+1) = \frac{x_j(t)-x_i(t)}{2} + 0.5 s_j(t) + \frac{s_i(t)}{2\delta_{out,i}(t)} \end{cases} \tag{5}$$

algoritmo con perdita di pacchetto:

$$\begin{cases} x_j(t+1) = \frac{x_j(t)+s_j(t)}{2} \\ s_j(t+1) = \frac{x_j(t)+s_j(t)}{2} \end{cases} \tag{6}$$

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

AGB

- Caso inconsapevole: l'aggiornamento dei nodi riceventi avviene allo stesso modo dei nodi isolati:

$$\begin{cases} x_j(t+1) = x_j(t) \\ s_j(t+1) = s_j(t) \end{cases} \tag{7}$$

In entrambi i casi se non si perde informazione anche l'aggiornamento dei nodi riceventi avviene come nell'algoritmo originale.

AG

Per implementare la perdita di pacchetto ci si serve, come già fatto nell'algoritmo AS, di variabili aleatorie binomiali $p_{ij}(t)$, che rappresentano la probabilità che la trasmissione dal nodo i al nodo j abbia avuto successo. L'aggiornamento del nodo che trasmette e dei nodi inattivi avviene come nell'algoritmo originale, mentre cambia l'aggiornamento del nodo ricevente:

- Algoritmo originale:

$$x_j(t+1) = x_j(t) + w_{ji}(x_i(t) - x_j(t)) + \epsilon w_{ji} s_j(t) \tag{8}$$

- Algoritmo con perdita di pacchetto:

$$x_j(t+1) = x_j(t) + w_{ji} p_{ij}(t)(x_i(t) - x_j(t)) + \epsilon w_{ji} s_j(t) \tag{9}$$

Simulazioni

Errore di convergenza

$$errore = \frac{1}{n} \left\| \mathbf{1}^T x(0) - \mathbf{1}^T x(\infty) \right\|_2$$

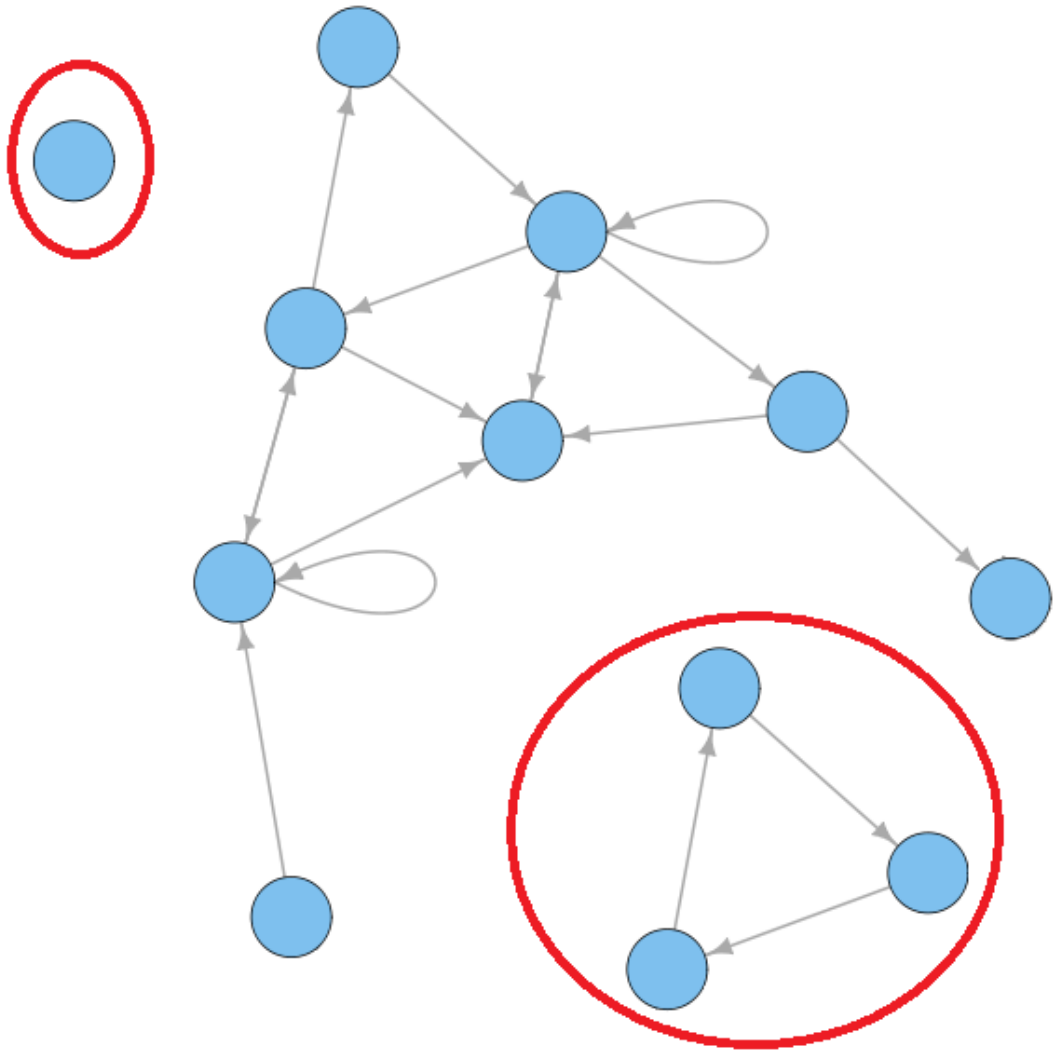
Tempo di Convergenza

$$\left\| x(t) - \frac{\mathbf{1}\mathbf{1}^T}{n} x(t) \right\|_2 < 0.5$$

Random Geometric Graph

- Gli n nodi sono collocati a caso in modo uniforme e indipendente.
- ogni coppia di nodi è collegata da un arco se :

$$d(u, v) < r$$



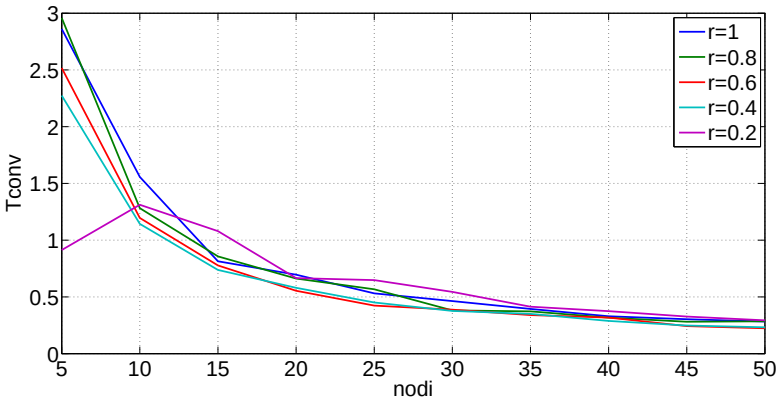
Algoritmi Sincroni

Errore di convergenza algoritmi sincroni con $r = 1$ e $p = 0.4$:

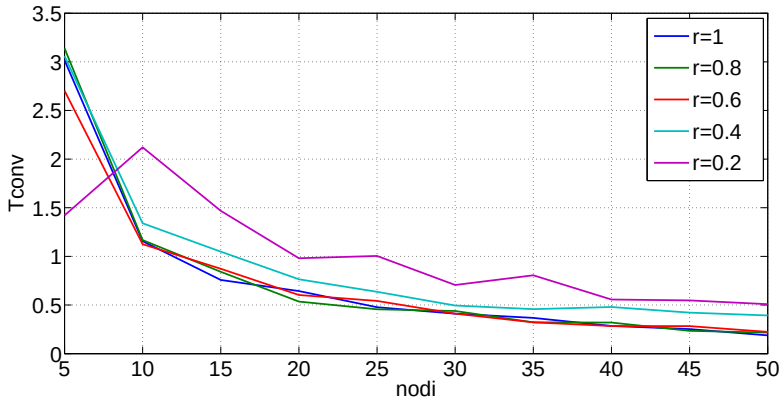
Nodi	ASP parziale	ASP bilanciato	AS
5	4.227	9.19863	50.8068
10	1.96352	4.07714	50.1957
15	1.24001	2.54749	50.1164
20	0.953461	1.52494	50.2019
25	0.885371	1.27471	49.4835
30	0.70758	0.988101	50.0245
35	0.567842	0.977046	49.6724
40	0.527323	0.707667	49.9628
45	0.436022	0.677762	49.6241
50	0.405194	0.631278	49.9521

ASP

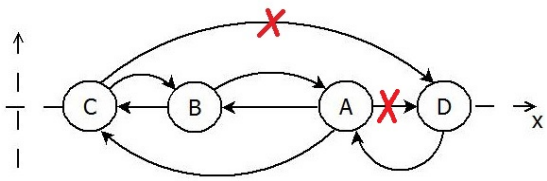
Algoritmi ASP con $p = 0.8$



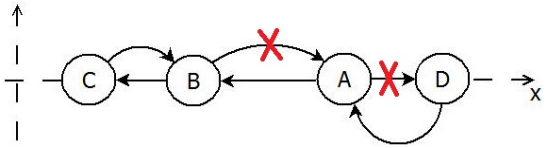
(a) Errore ASP parziale



(b) Errore ASP bilanciato



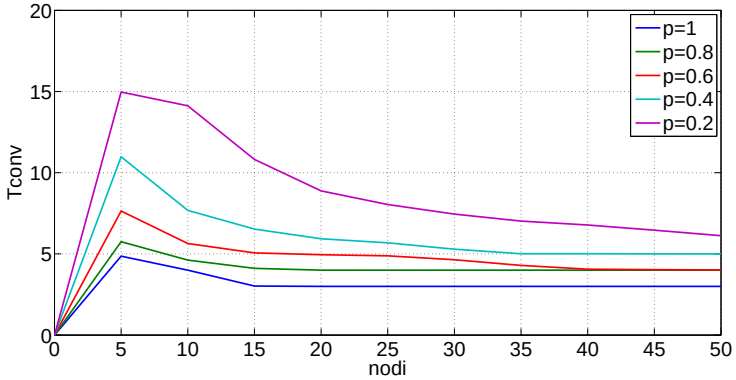
(c)



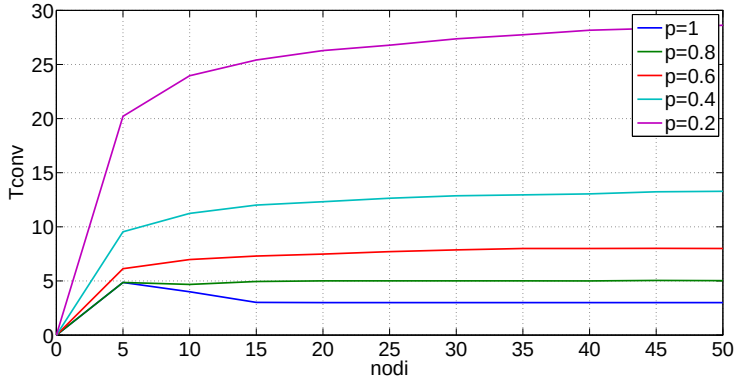
(d)

ASP

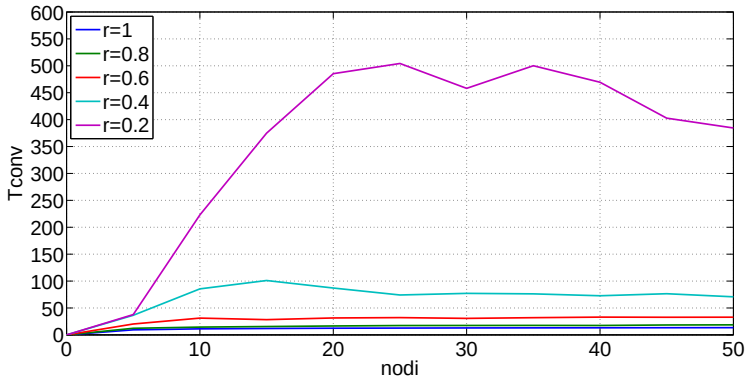
ASP con $r = 1$ (sopra) e ASP con $p = 0.4$ (sotto)



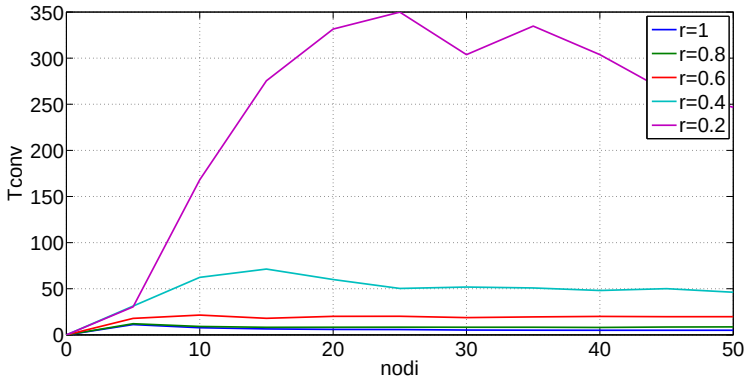
(e) T_{conv} ASP parziale



(f) T_{conv} ASP bilanciato



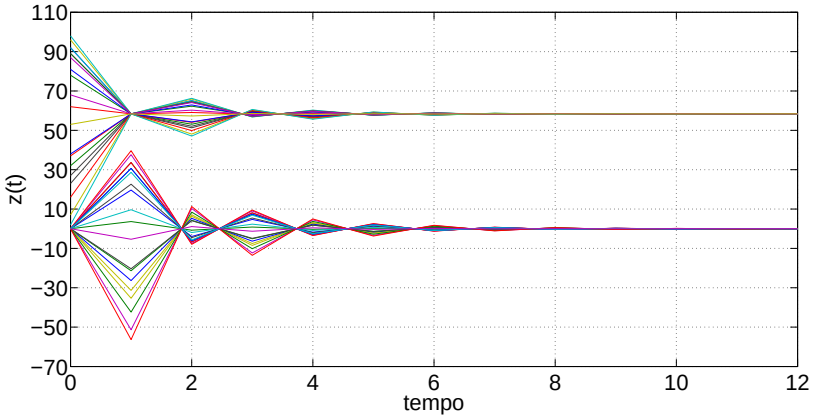
(g)

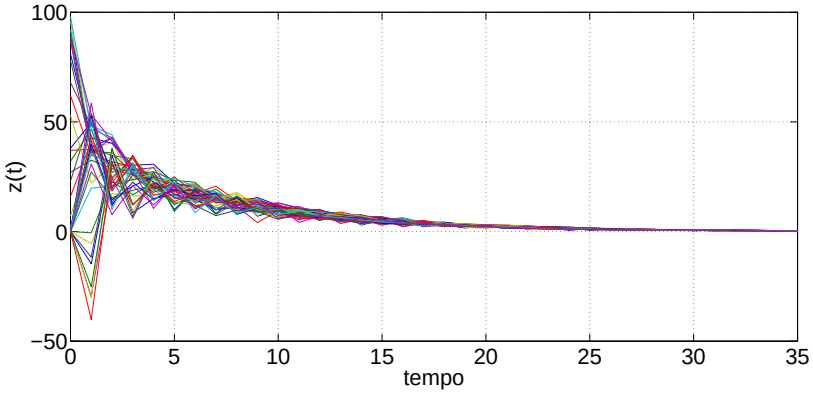


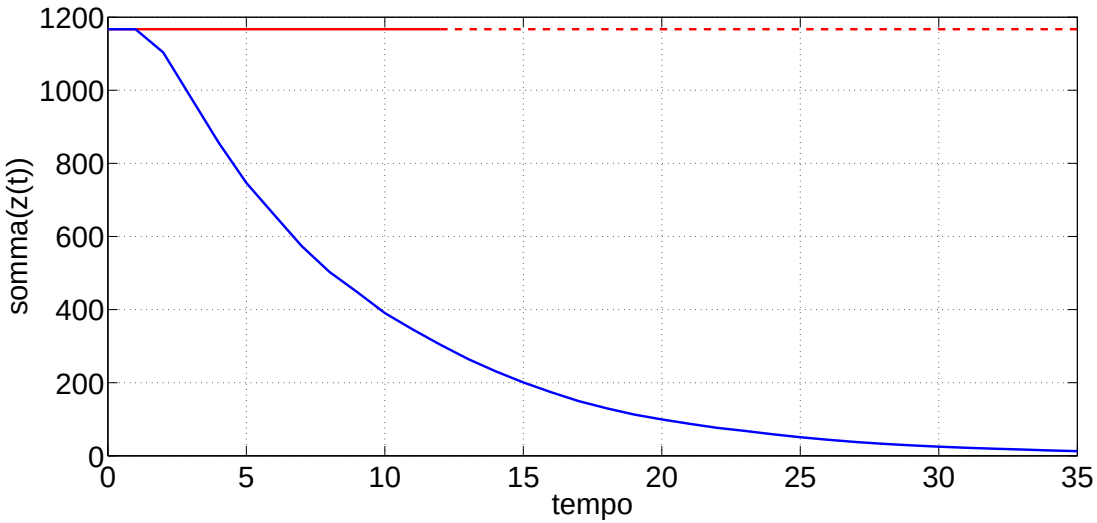
(h)

AS

Grafo con $n = 20$, $p = 0.8$: e $r = 1$

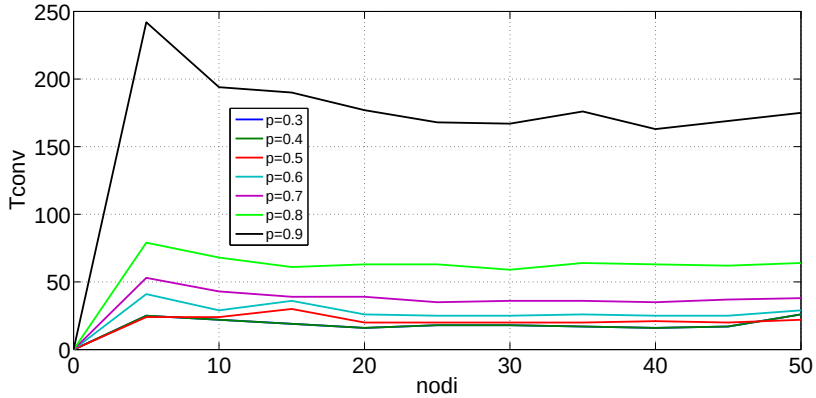




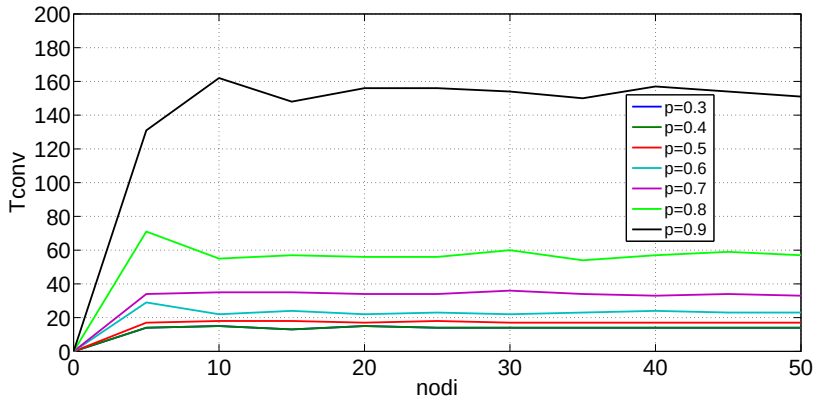


AS

Velocità di convergenza:



(i) $r = 0.5$



(j) $r = 1$

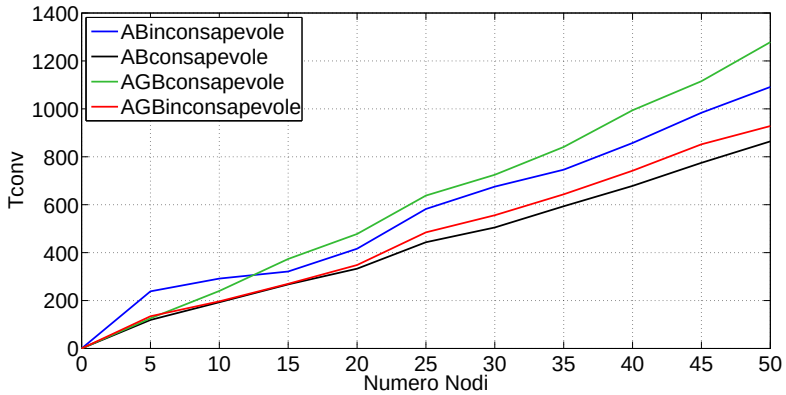
Algoritmi Asincroni

Si comincia confrontando l'errore di convergenza degli algoritmi broadcast con $r = 0.6$ e $p = 0.3$:

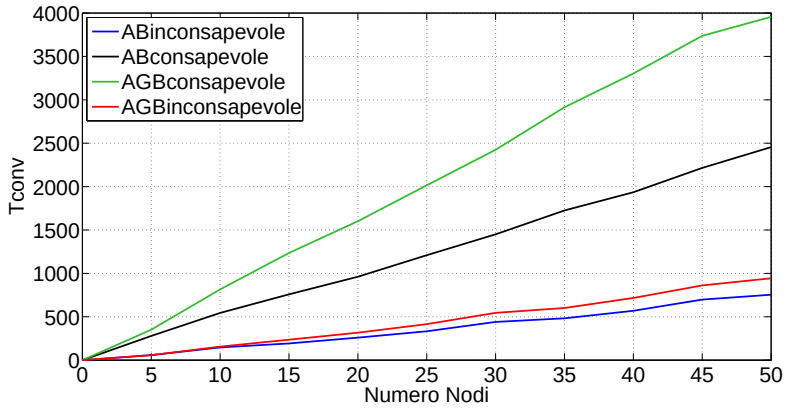
Nodi	AB incons	AB cons	AGB cons	AGB incons
5	3.83925	36.6317	36.5767	5.00535
10	5.0101	49.6073	49.5928	6.11589
15	3.13393	64.9368	64.9254	3.59227
20	2.18521	39.998	39.9868	2.62573
25	1.96511	44.1988	44.1857	2.04788
30	1.88671	49.7668	49.7552	2.31478
35	2.0325	44.2226	44.216	2.36736
40	1.94613	49.9449	49.9323	2.33459
45	1.48397	42.3727	42.3618	1.48008
50	1.97658	45.4146	45.4018	2.33264

Algoritmi Asincroni

Velocità di convergenza algoritmi broadcast con $r = 0.9$:



(k) $p = 0.3$

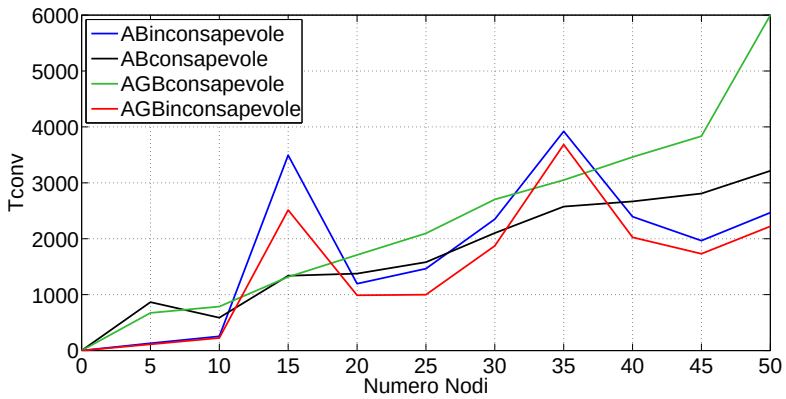


(l) $p = 0.7$

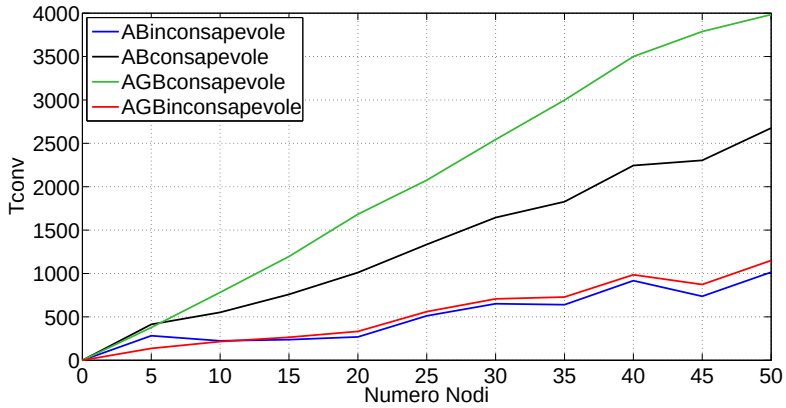
La velocità di convergenza degli algoritmi consapevoli diminuisce all'aumentare della probabilità di successo.

Algoritmi Asincroni

Velocità di convergenza algoritmi broadcast con $p = 0.7$:



(m) $r = 0.4$

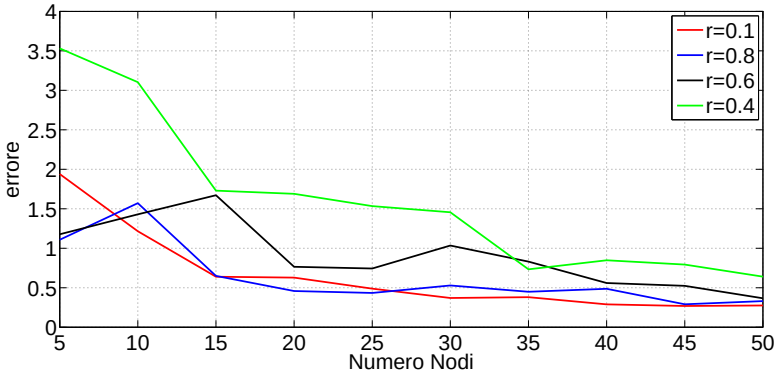


(n) $r = 0.7$

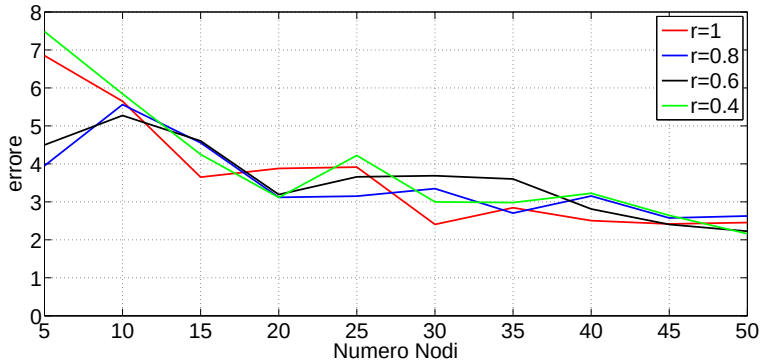
L'AGB inconsapevole ha velocità di convergenza maggiore rispetto all'AB inconsapevole nel caso di reti con raggio di vicinanza basso, mentre si verifica il contrario per grafi con raggio di vicinanza più elevato.

Algoritmi Asincroni

Confronto algoritmi broadcast inconsapevoli con AG.
Errore di convergenza ($p = 0.8$):



(o) AGB inconsapevole

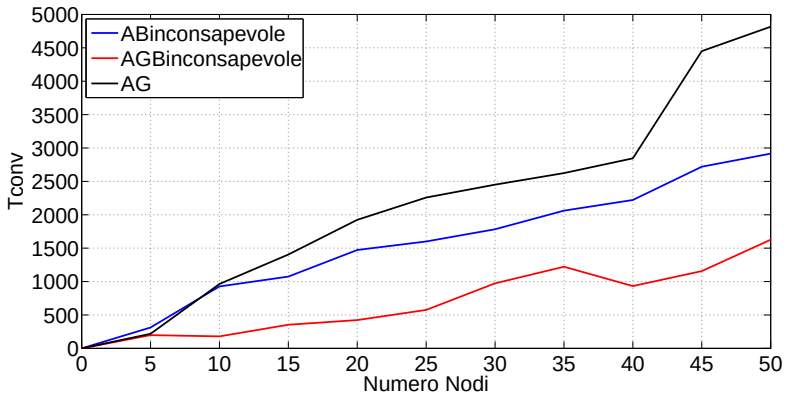


(p) AG

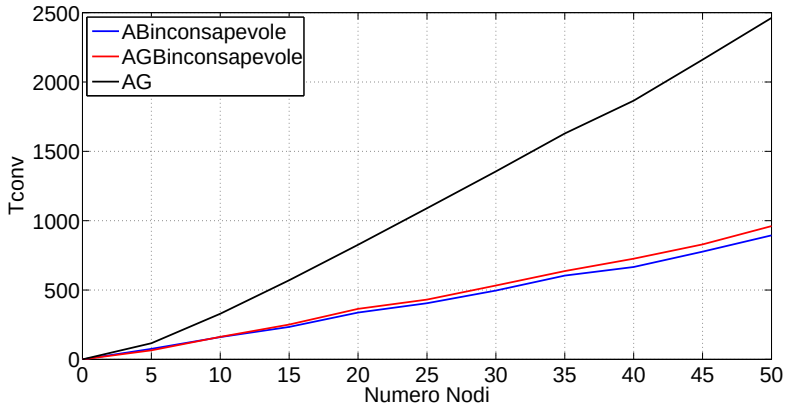
L'errore di convergenza negli algoritmi broadcast ha un andamento inversamente proporzionale al numero di agenti della rete e al raggio di vicinanza; tale comportamento non si ravvisa in AG.

Algoritmi Asincroni

Velocità di convergenza ($p = 0.5$):



(q) $r = 0.4$

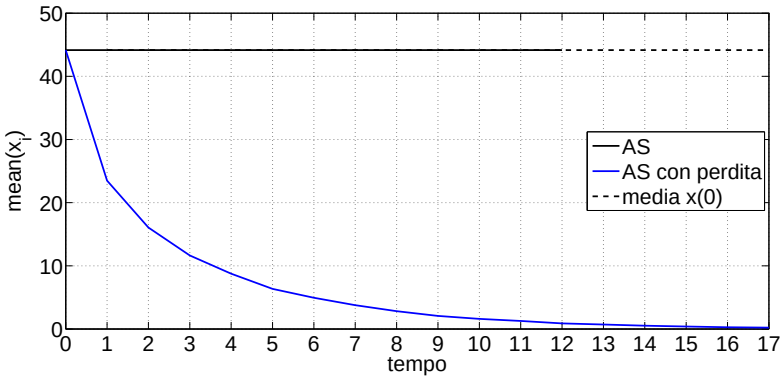


(r) $r = 0.9$

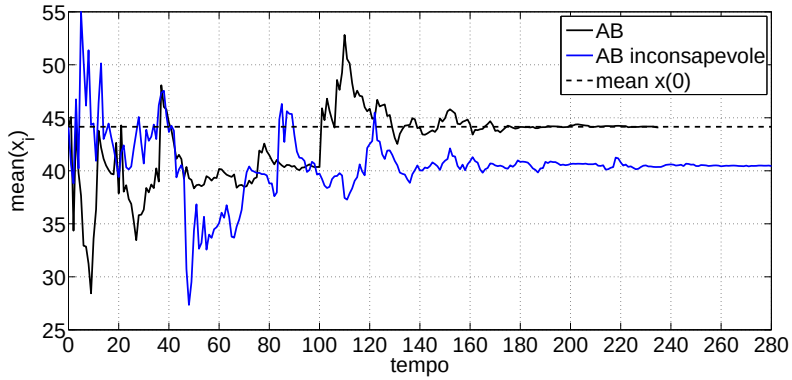
All'aumentare del raggio di vicinanza le prestazioni di AG peggiorano sensibilmente rispetto a quelle di AB e AGB inconsapevoli.

Algoritmi Asincroni

Confronto errori di convergenza AS e AB in un grafo con 20 nodi, $r = 1$ e $p = 0.5$:



(s) AS



(t) AB

Errore di convergenza di AS molto più elevato di AB.

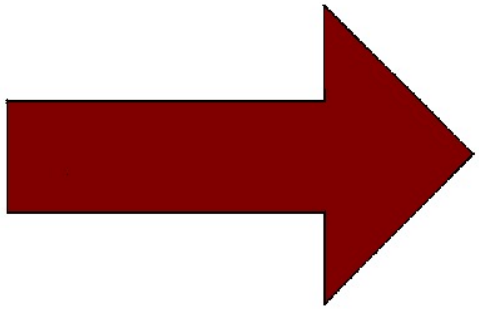
Algoritmi Asincroni

Errore di convergenza degli algoritmi ASP, AB inconsapevole, AGB inconsapevole e AG con $r = 0.4$ e $p = 0.8$:

Nodi	ASP parz	ASP bilanc	AB incons	AGB incons	AG
5	2.27447	3.05841	2.7847	3.53136	7.48396
10	1.14374	1.34029	2.4032	3.103	5.84223
15	0.737217	1.04853	1.54728	1.73018	4.24831
20	0.580298	0.764593	1.15068	1.68986	3.11462
25	0.450678	0.635754	1.37454	1.53367	4.22039
30	0.376165	0.494718	1.04942	1.45601	2.99758
35	0.349976	0.457199	0.822409	0.733045	2.98101
40	0.287984	0.479261	0.862267	0.847636	3.22506
45	0.247911	0.421956	0.789985	0.793587	2.64309
50	0.231825	0.392276	0.696223	0.640354	2.15904

Reti Sincrone

- Errore di convergenza: ASP molto meglio di AS
- Velocità di convergenza: prestazioni simili



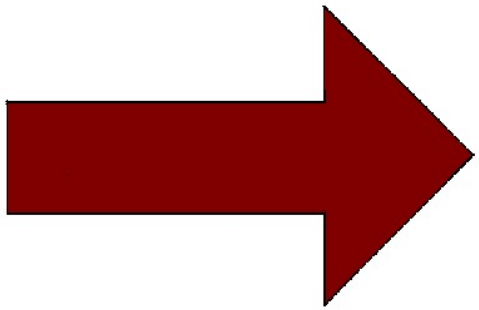
ASP meglio di AS

- Compensazione parziale leggermente meglio in termini di errore di convergenza
- Compensazione bilanciata meglio in termini di velocità di convergenza

Reti Asincrone

Algoritmi broadcast:

- Errore di convergenza: algoritmi inconsapevoli molto meglio dei consapevoli, leggera preferenza per AB inconsapevole



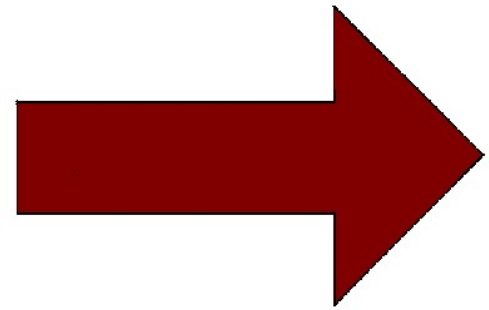
si tralasciano gli algoritmi consapevoli

- Velocità di convergenza:
  - AGB inconsapevole meglio in reti con r basso
  - AB inconsapevole meglio in reti con r alto

Reti Asincrone

Algoritmo gossip:

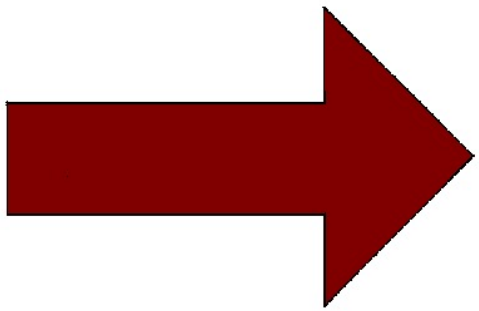
- Errore di convergenza: prestazioni vicine a quelle degli algoritmi broadcast inconsapevoli
- Velocità di convergenza:
  - prestazioni accettabili per reti con r basso
  - se r elevato c'è un divario maggiore rispetto agli algoritmi broadcast inconsapevoli



se possibile è meglio lavorare con gli algoritmi broadcast

Confronto algoritmi sincroni-asincroni

- Errore di convergenza: gli algoritmi broadcast inconsapevoli e AG hanno prestazioni molto vicine rispetto ad ASP
- Velocità di convergenza: con gli algoritmi sincroni si ottengono risultati decisamente migliori



la soluzione ideale è lavorare con algoritmi sincroni come ASP in cui si sostituisce l'informazione persa con i dati disponibili

Sviluppi Futuri

- Introdurre la perdita di pacchetto in modo diverso a quanto fatto in questo progetto
- Prendere in considerazioni anche altri algoritmi
- Trovare nuovi criteri per valutare gli algoritmi
- In sede di simulazione utilizzare anche altri tipi di grafi oltre i random geometric graph
- Integrare i nuovi risultati a quelli presenti in questo progetto
- Cercare di sviluppare una teoria scientifica sull'average consensus con perdita di pacchetto