

```
addpath('lib')
addpath('lib\funzioni')
addpath('lib\Sottoclassi_di_Distributions')
rng(343310+341965)
```

Bello Renato - 341965, Chiodo Martina - 343310

## HOMework BA

### Introduzione

Per questo homework abbiamo sviluppato una libreria base per la simulazione di un processo modellabile tramite una **Rete di Jackson**, composta da un insieme di code e server. I principali oggetti della libreria sono:

- **Network**: rappresenta l'intera rete in cui si svolge la simulazione. È composta da un insieme di nodi collegati tra loro secondo quanto specificato da una **matrice di adiacenza stocastica**, che definisce le probabilità di transizione tra i nodi. Si noti che la matrice di adiacenza è da definire in modo tale che esista un **nodo sink**, ossia un nodo attrattore raggiungibile da qualsiasi altro nodo della rete e da cui non è possibile uscire. All'interno del costruttore vengono eseguiti alcuni controlli per verificare che la matrice di adiacenza fornita soddisfi determinate proprietà. In primo luogo, la matrice deve essere quadrata. In secondo luogo, deve contenere un solo nodo sink, cioè un nodo che non ha uscite. Infine, da ciascun nodo in cui è definita una distribuzione di arrivo deve esistere almeno un percorso che conduca al nodo sink. L'ultimo controllo viene effettuato utilizzando l'algoritmo *Breadth-First Search* (BFS), che permette di determinare i nodi raggiungibili a partire da un nodo dato. Questo controllo è efficace solo nel caso in cui il grafo non contenga archi condizionati, ovvero quando tutti gli archi sono definiti da valori numerici e non da *function handle*. In presenza di archi rappresentati da *function handle*, ossia con condizioni che dipendono dagli attributi delle entità, l'algoritmo, per come è stato implementato, non è in grado di determinare se l'entità è in grado di raggiungere effettivamente il nodo sink. Potrebbero infatti esistere combinazioni di attributi che causano cicli infiniti o impediscono il raggiungimento del sink.
- **Nodo**: è l'elemento base della rete. Ogni nodo è costituita da una **coda**, che può essere gestita con politica *FIFO* o *LIFO*, da un **insieme di server**, ciascuno caratterizzato da una sua distribuzione di tempi di servizio, e da una **distribuzione degli interarrivi** per modellare l'arrivo di entità esterne alla rete nel determinato nodo.
- **Event**: rappresenta il meccanismo che fa avanzare la simulazione. A ogni evento corrisponde un'azione specifica che modifica lo stato del sistema. Nella libreria, questo concetto è modellato tramite una classe astratta, dalla quale derivano classi specifiche per ciascun tipo di evento. In particolare, nel nostro caso le classi derivate sono **ArrivalEvent** ed **EndProcessEvent**, che rappresentano rispettivamente l'arrivo di un'entità in un nodo e la fine del servizio di un'entità.
- **Simulation**: rappresenta il motore centrale della libreria. Al suo interno vengono aggregate tutte le informazioni relative alla rete. Questa classe fornisce un metodo dedicato per l'avvio e la gestione della simulazione, che si occupa di eseguire gli eventi nell'ordine corretto e di aggiornare lo stato del sistema nel tempo.
- **Distribution**: classe che contiene tutte le informazioni relative alle distribuzioni di probabilità utili nella libreria.

Inoltre, nella libreria vi sono delle classi che gestiscono le **statistiche** della simulazione, noi abbiamo scelto di considerare sia delle statistiche globali che delle statistiche legate ad una singola coda. In particolare le statistiche da noi implementate sono:

- il *numero medio di entità nel sistema*, salvata nella classe **AverageEntityIntoNetwork**
- il *tempo medio che ogni entità impiega ad arrivare al nodo sink*, salvata nella classe **AverageEntityNetworkTime**
- il *numero medio di entità in una specifica coda*, salvata nella classe **AverageNumEntityIntoNodes**
- il *tempo medio di attesa in una specifica coda*, salvata nella classe **AverageEntityQueuesTime**

## Esempio di utilizzo

Per utilizzare la libreria è innanzitutto necessario definire la rete

Ciò viene fatto definendo **una matrice di adiacenza** che rappresenta i possibili percorsi all'interno del sistema. Il cammino seguito dalle entità non è univoco, ma può essere determinato in modo probabilistico oppure in base a proprietà specifiche dell'entità. Quest'ultimo approccio è reso possibile attraverso l'inclusione di *function handle* all'interno della matrice, che consentono di selezionare dinamicamente il nodo successivo. È importante assicurarsi che la rete includa un nodo globalmente attrattivo, che agisca come **sink** del sistema, ossia come nodo finale in cui le entità terminano il loro percorso.

```
matrice_adiacenza = {  
    0, 0.4, 0.6, 0, 0, 0;  
    0, 0, 0, @(ent) ent.prop == 1, @(ent) ent.prop == 0, 0;  
    0, 0, 0, 0, 0, 1;  
    0.2, 0, 0, 0, 0, 0.8;  
    0, 0, 0, 0, 0, 1;  
    0, 0, 0, 0, 0, 1; % il nodo 6 è il sink  
};  
  
n = size(matrice_adiacenza, 1);
```

Successivamente, è necessario definire le **distribuzioni degli interarrivi** e dei **tempi di servizio** e le **policy delle code**. Questo viene fatto utilizzando dei *cell array* di lunghezza pari al numero di nodi della rete. In particolare, per i tempi di servizio, ogni nodo è associato a un *cell array* contenente un numero di elementi pari al numero di server presenti nel nodo. Gli elementi di questi *cell array* sono oggetti che rappresentano distribuzioni di probabilità, definite all'interno della libreria.

```
% Una distribuzione di interarrivo per ciascun nodo: il -1 è un valore  
% usato per dire che il corrispondente nodo non supporta arrivi esterni.  
distr_InterArrivi = cell(n,1);  
distr_InterArrivi{1} = Exponential(13);  
distr_InterArrivi{2} = Exponential(8);  
distr_InterArrivi{3} = -1;  
distr_InterArrivi{4} = -1;  
distr_InterArrivi{5} = Uniform(10,10,true);  
distr_InterArrivi{6} = -1;  
  
% Un cell array per ciascun nodo con distribuzioni di servizio per ciascun  
% server del nodo.  
distr_TempiServizio = cell(n,1);
```

```
distr_TempiServizio{1} = {Exponential(30), Exponential(30)};
distr_TempiServizio{2} = {Exponential(10)};
distr_TempiServizio{3} = {Exponential(18)};
distr_TempiServizio{4} = {Exponential(15)};
distr_TempiServizio{5} = {Uniform(5,20,false)};
distr_TempiServizio{6} = {Uniform(inf, inf, false)};
```

% Una cell array per definire la policy della coda di ciascun nodo.

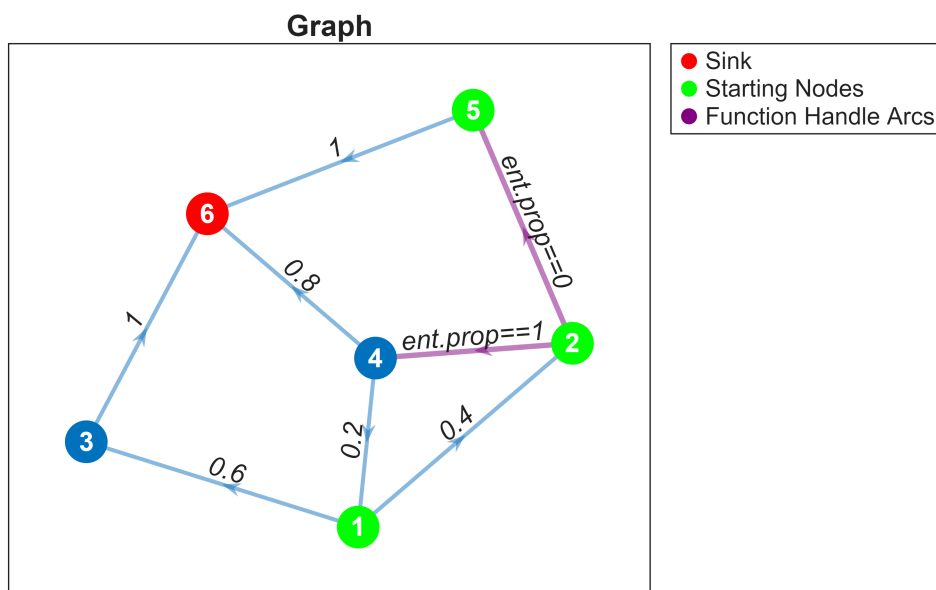
```
policy = cell(n,1);
policy{1} = 'FIFO';
policy{2} = 'FIFO';
policy{3} = 'LIFO';
policy{4} = 'FIFO';
policy{5} = 'FIFO';
policy{6} = 'FIFO';
```

Si osservi che al nodo sink è associato un server con tempo di servizio infinito, poiché esso non rappresenta un vero nodo della rete, ma rappresenta semplicemente un segnaposto per indicare il termine del percorso.

Tutte queste informazioni vengono passate alla classe **Network** che si occuperà di inizializzare la rete.

% Costruzione dell'oggetto Network

```
net = Network(matrice_adiacenza, distr_InterArrivi, distr_TempiServizio, policy);
stampa_grafo(net,true)
```



La simulazione vera e propria viene inizializzata passando al costruttore della classe **Simulation** l'oggetto Network precedentemente definito eventualmente insieme a una lista di attributi con i relativi valori possibili per le entità. In questo modo, a ogni entità generata durante la simulazione viene assegnato casualmente uno dei valori specificati, rendendo possibile la personalizzazione del comportamento in base alle sue caratteristiche.

% Inizializzo una simulazione

```
attributi_entita = [];
```

```
attributi_entita.prop = {0, 1};
sim = Simulation(net, attributi_entita);
```

A questo punto, il sistema è pronto per eseguire la simulazione, avviata tramite il metodo `run()` della classe `Simulation`. Questo metodo richiede in input l'orizzonte temporale, che definisce il criterio di terminazione della simulazione, e un parametro booleano (`verbose`) che regola il livello di dettaglio dell'output prodotto durante l'esecuzione.

```
TIMEORIZON = 100;
verbose = true;
stats = sim.run(TIMEORIZON, verbose);
```

```
*****
INIZIO SIMULAZIONE ...
CLOCK = 1.58
L'entità 1 è stata generata al nodo 1.
L'entità 1 ha iniziato il servizio presso il server 1 del nodo 1.

CLOCK = 5.61
L'entità 2 è stata generata al nodo 2.
L'entità 2 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 8.10
L'entità 3 è stata generata al nodo 1.
L'entità 3 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 8.64
L'entità 4 è stata generata al nodo 2.

CLOCK = 10.00
L'entità 5 è stata generata al nodo 5.
L'entità 5 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 13.32
L'entità 6 è stata generata al nodo 2.

CLOCK = 14.91
L'entità 7 è stata generata al nodo 1.

CLOCK = 15.66
L'entità 2 ha terminato il servizio al nodo 2 ed è passata al nodo 5.
L'entità 4 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 18.77
L'entità 1 ha terminato il servizio al nodo 1 ed è passata al nodo 2.
L'entità 7 ha iniziato il servizio presso il server 1 del nodo 1.

CLOCK = 19.15
L'entità 8 è stata generata al nodo 1.

CLOCK = 20.00
L'entità 9 è stata generata al nodo 5.

CLOCK = 23.79
L'entità 4 ha terminato il servizio al nodo 2 ed è passata al nodo 4.
L'entità 4 ha iniziato il servizio presso il server 1 del nodo 4.
L'entità 6 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 29.22
L'entità 5 ha terminato il servizio al nodo 5 ed è passata al nodo 6.
L'entità 2 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 30.00
L'entità 10 è stata generata al nodo 5.
```

CLOCK = 31.13  
L'entità 7 ha terminato il servizio al nodo 1 ed è passata al nodo 3.  
L'entità 7 ha iniziato il servizio presso il server 1 del nodo 3.  
L'entità 8 ha iniziato il servizio presso il server 1 del nodo 1.

CLOCK = 37.33  
L'entità 11 è stata generata al nodo 2.

CLOCK = 37.70  
L'entità 6 ha terminato il servizio al nodo 2 ed è passata al nodo 5.  
L'entità 1 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 40.00  
L'entità 12 è stata generata al nodo 5.

CLOCK = 40.17  
L'entità 1 ha terminato il servizio al nodo 2 ed è passata al nodo 5.  
L'entità 11 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 40.77  
L'entità 13 è stata generata al nodo 2.

CLOCK = 42.97  
L'entità 14 è stata generata al nodo 1.

CLOCK = 45.99  
L'entità 2 ha terminato il servizio al nodo 5 ed è passata al nodo 6.  
L'entità 9 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 46.15  
L'entità 15 è stata generata al nodo 1.

CLOCK = 48.41  
L'entità 7 ha terminato il servizio al nodo 3 ed è passata al nodo 6.

CLOCK = 48.93  
L'entità 11 ha terminato il servizio al nodo 2 ed è passata al nodo 4.  
L'entità 13 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 50.00  
L'entità 16 è stata generata al nodo 5.

CLOCK = 51.88  
L'entità 17 è stata generata al nodo 1.

CLOCK = 54.00  
L'entità 18 è stata generata al nodo 1.

CLOCK = 54.05  
L'entità 13 ha terminato il servizio al nodo 2 ed è passata al nodo 5.

CLOCK = 55.96  
L'entità 19 è stata generata al nodo 2.  
L'entità 19 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 57.72  
L'entità 9 ha terminato il servizio al nodo 5 ed è passata al nodo 6.  
L'entità 10 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 58.15  
L'entità 20 è stata generata al nodo 2.

CLOCK = 58.82  
L'entità 3 ha terminato il servizio al nodo 1 ed è passata al nodo 3.  
L'entità 3 ha iniziato il servizio presso il server 1 del nodo 3.  
L'entità 14 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 59.25  
L'entità 3 ha terminato il servizio al nodo 3 ed è passata al nodo 6.

CLOCK = 60.00  
L'entità 21 è stata generata al nodo 5.

CLOCK = 69.17  
L'entità 14 ha terminato il servizio al nodo 1 ed è passata al nodo 3.  
L'entità 14 ha iniziato il servizio presso il server 1 del nodo 3.  
L'entità 15 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 70.00  
L'entità 22 è stata generata al nodo 5.

CLOCK = 70.60  
L'entità 23 è stata generata al nodo 1.

CLOCK = 70.67  
L'entità 24 è stata generata al nodo 2.

CLOCK = 70.88  
L'entità 19 ha terminato il servizio al nodo 2 ed è passata al nodo 5.  
L'entità 20 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 71.49  
L'entità 8 ha terminato il servizio al nodo 1 ed è passata al nodo 2.  
L'entità 17 ha iniziato il servizio presso il server 1 del nodo 1.

CLOCK = 73.01  
L'entità 4 ha terminato il servizio al nodo 4 ed è passata al nodo 6.  
L'entità 11 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 73.52  
L'entità 15 ha terminato il servizio al nodo 1 ed è passata al nodo 3.  
L'entità 18 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 74.67  
L'entità 20 ha terminato il servizio al nodo 2 ed è passata al nodo 5.  
L'entità 24 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 75.36  
L'entità 24 ha terminato il servizio al nodo 2 ed è passata al nodo 5.  
L'entità 8 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 76.48  
L'entità 25 è stata generata al nodo 2.

CLOCK = 77.05  
L'entità 10 ha terminato il servizio al nodo 5 ed è passata al nodo 6.  
L'entità 6 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 78.38  
L'entità 8 ha terminato il servizio al nodo 2 ed è passata al nodo 5.  
L'entità 25 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 80.00  
L'entità 26 è stata generata al nodo 5.

CLOCK = 82.16  
L'entità 6 ha terminato il servizio al nodo 5 ed è passata al nodo 6.  
L'entità 12 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 83.67  
L'entità 25 ha terminato il servizio al nodo 2 ed è passata al nodo 5.

CLOCK = 83.73  
L'entità 14 ha terminato il servizio al nodo 3 ed è passata al nodo 6.  
L'entità 15 ha iniziato il servizio presso il server 1 del nodo 3.

CLOCK = 85.74  
L'entità 15 ha terminato il servizio al nodo 3 ed è passata al nodo 6.

CLOCK = 86.90  
L'entità 27 è stata generata al nodo 1.

CLOCK = 89.13  
L'entità 28 è stata generata al nodo 2.  
L'entità 28 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 89.19  
L'entità 28 ha terminato il servizio al nodo 2 ed è passata al nodo 4.

CLOCK = 90.00  
L'entità 29 è stata generata al nodo 5.

CLOCK = 91.39  
L'entità 30 è stata generata al nodo 2.  
L'entità 30 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 91.90  
L'entità 31 è stata generata al nodo 2.

CLOCK = 93.31  
L'entità 12 ha terminato il servizio al nodo 5 ed è passata al nodo 6.  
L'entità 1 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 100.00  
L'entità 32 è stata generata al nodo 5.

FINE SIMULAZIONE

Avarage time into the network per entity = 45.08

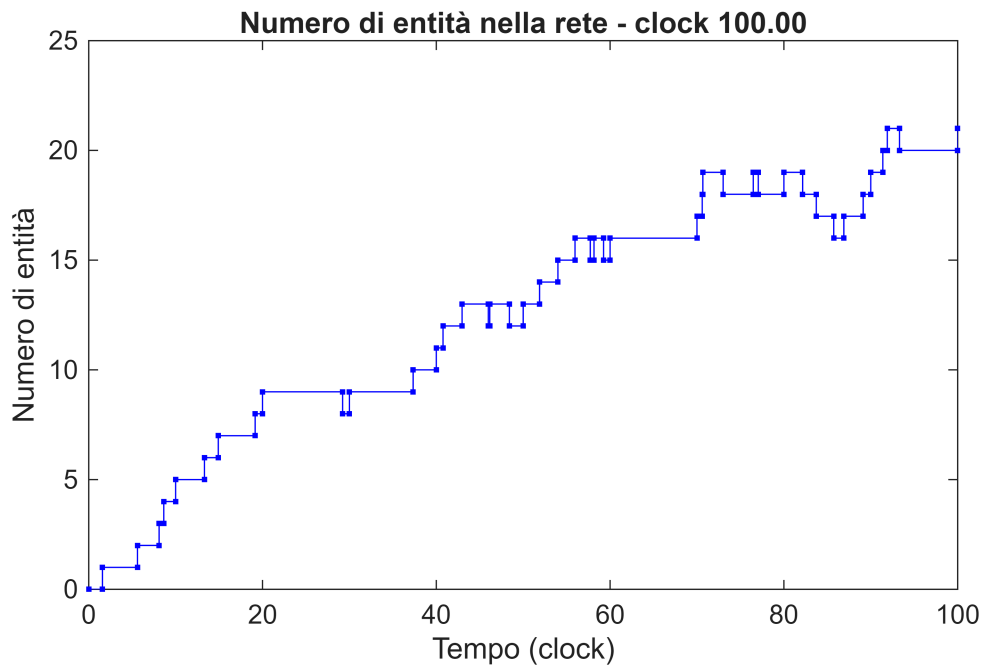
Avarage time spent in the queues per entity:

Node 1: 25.14  
Node 2: 11.56  
Node 3: 2.55  
Node 4: 37.76  
Node 5: 57.31  
Node 6: SINK

Avarage number of entities in the network = 12.50

Avarage number of entities in the nodes of the network:

Node 1: 1.36  
Node 2: 0.78  
Node 3: 0.10  
Node 4: 0.35  
Node 5: 5.15  
Node 6: SINK



Il metodo `run()` restituisce un *cell array* contenente tutte le statistiche raccolte durante la simulazione, utili per l'analisi delle prestazioni del sistema simulato. Tali statistiche possono essere visualizzate direttamente durante l'esecuzione se il parametro `verbose` è impostato a `true`, oppure consultate successivamente accedendo al *cell array* restituito in output.

Nelle prossime righe di codice seguiranno delle stampe esemplificative di come accedere alle statistiche.

`stats`

`stats = 1×4 cell`

...

	1	2	3
1	1×1 AverageEntityNetworkTime	1×1 AverageEntityQueuesTime	1×1 AverageNumEntityIntoNetwork

```
fprintf("Il flow time medio per ogni entità è pari a %.2f.", stats{1}.results)
```

Il flow time medio per ogni entità è pari a 45.08.

```
fprintf("Il tempo medio complessivo che un'entità passa nella coda associata al nodo 3 è pari a %.2f.", stats{2}.results(3))
```

Il tempo medio complessivo che un'entità passa nella coda associata al nodo 3 è pari a 2.55.

```
fprintf("Il numero medio di entità nella rete è pari a %.2f.", stats{3}.results)
```

Il numero medio di entità nella rete è pari a 12.50.

```
fprintf("Il numero medio di entità presenti nella coda del nodo 5 è pari a %.2f.", stats{4}.results(5))
```

Il numero medio di entità presenti nella coda del nodo 5 è pari a 5.15.



Crediamo che l'analisi delle statistiche per singolo nodo sia utile per individuare i colli di bottiglia e guidare interventi di **ottimizzazione** della rete. Ad esempio, se il nodo 5 presenta i tempi di attesa più elevati, l'aggiunta di un server potrebbe contribuire a un'ottimizzazione complessiva del flow time medio del sistema.

```
distr_TempiServizio{5} = {Uniform(5,20,false), Uniform(5,20,false)};
net = Network(matrice_adiacenza, distr_InterArrivi, distr_TempiServizio, policy);
sim = Simulation(net, attributi_entita);
stats = sim.run(TIMEORIZON, false);
```

```
*****
INIZIO SIMULAZIONE ...
FINE SIMULAZIONE
```

```
fprintf("Il flow time medio per ogni entità è pari a %.2f.", stats{1}.results)
```

Il flow time medio per ogni entità è pari a 33.75.

## ESEMPIO DI RETE BIPARTITA

Vediamo un secondo esempio di applicazione con una rete leggermente più grande.

```
matrice_adiacenza = {
% 1    2    3    4    5    6    7    8    9    10   11   12   13
14    15
    0    0    0    0  0.25  0.25  0.25  0.25  0    0    0    0    0
0    0; % nodo 1
    0    0    0    0  0.25  0.25  0.25  0.25  0    0    0    0    0
0    0; % nodo 2
    0    0    0    0  0.25  0.25  0.25  0.25  0    0    0    0    0
0    0; % nodo 3
    0    0    0    0  0.25  0.25  0.25  0.25  0    0    0    0    0
0    0; % nodo 4

    0    0    0    0    0    0    0    0  0.25  0.25  0.25  0.25  0
0    0; % nodo 5
    0    0    0    0    0    0    0    0  0.25  0.25  0.25  0.25  0
0    0; % nodo 6
    0    0    0    0    0    0    0    0  0.25  0.25  0.25  0.25  0
0    0; % nodo 7
    0    0    0    0    0    0    0    0  0.25  0.25  0.25  0.25  0
0    0; % nodo 8

    0    0    0    0    0    0    0    0    0    0    0    0  0.5 0.5 0; %
nodo 9
    0    0    0    0    0    0    0    0    0    0    0    0  0.5 0.5 0; %
nodo 10
    0    0    0    0    0    0    0    0    0    0    0    0  0.5 0.5 0; %
nodo 11
    0    0    0    0    0    0    0    0    0    0    0    0  0.5 0.5 0; %
nodo 12

    0    0    0    0    0    0    0    0    0    0    0    0    0
0    1; % nodo 13 output
```

```

0 0 0 0 0 0 0 0 0 0 0 0 0
0 1; % nodo 14 output
0 0 0 0 0 0 0 0 0 0 0 0 0
0 1; % nodo 15 output
};

n = size(matrice_adiacenza, 1);

% Distribuzione interarrivi: nuovi nodi intermedi non hanno arrivi esterni
distr_InterArrivi = cell(n,1);
distr_InterArrivi{1} = Exponential(20);
distr_InterArrivi{2} = Exponential(15);
distr_InterArrivi{3} = Exponential(35);
distr_InterArrivi{4} = Exponential(5);
for i = 5:n
    distr_InterArrivi{i} = -1; % nessun arrivo esterno
end

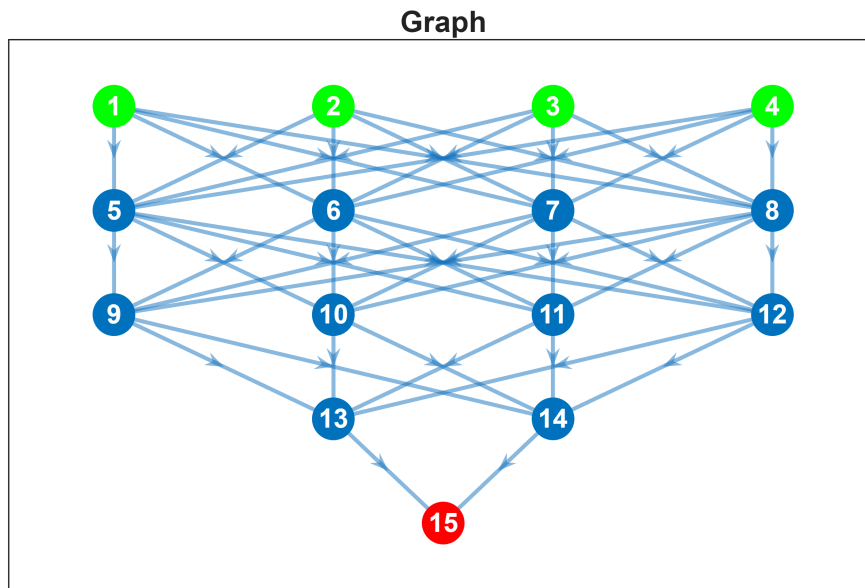
% Distribuzione tempi di servizio per ogni nodo (usiamo analoghe distribuzioni,
o varianti)
distr_TempiServizio = cell(n,1);
distr_TempiServizio{1} = {Exponential(30), Exponential(30)};
distr_TempiServizio{2} = {Exponential(2)};
distr_TempiServizio{3} = {Exponential(18)};
distr_TempiServizio{4} = {Exponential(1)};
distr_TempiServizio{5} = {Uniform(5,20,false)};
distr_TempiServizio{6} = {Exponential(18)};
distr_TempiServizio{7} = {Exponential(18)};
distr_TempiServizio{8} = {Uniform(5,20,false)};
distr_TempiServizio{9} = {Exponential(30), Exponential(30)};
distr_TempiServizio{10} = {Uniform(5,20,false)};
distr_TempiServizio{11} = {Exponential(10)};
distr_TempiServizio{12} = {Exponential(12)};
distr_TempiServizio{13} = {Uniform(3,15,false)};
distr_TempiServizio{14} = {Exponential(20)};
distr_TempiServizio{15} = {Uniform(inf, inf, false)};

% Policy code queue per ogni nodo
policy = cell(n,1);
policy{1} = 'FIFO';
policy{2} = 'FIFO';
policy{3} = 'LIFO';
policy{4} = 'FIFO';
policy{5} = 'FIFO';
policy{6} = 'FIFO';
policy{7} = 'LIFO';
policy{8} = 'FIFO';
policy{9} = 'FIFO';
policy{10} = 'FIFO';
policy{11} = 'FIFO';
policy{12} = 'LIFO';
policy{13} = 'FIFO';
policy{14} = 'FIFO';
policy{15} = 'FIFO';

```

```
% Costruzione dell'oggetto Network
```

```
net = Network(matrice_adiacenza, distr_InterArrivi, distr_TempiServizio, policy);  
stampa_grafo(net, false);
```



```
% Inizializzo una simulazione
```

```
sim = Simulation(net);  
TIMEORIZON = 100;  
stats2 = sim.run(TIMEORIZON,true);
```

```
*****
```

```
INIZIO SIMULAZIONE ...
```

```
CLOCK = 0.67
```

```
L'entità 1 è stata generata al nodo 4.
```

```
L'entità 1 ha iniziato il servizio presso il server 1 del nodo 4.
```

```
CLOCK = 0.81
```

```
L'entità 2 è stata generata al nodo 4.
```

```
CLOCK = 0.98
```

```
L'entità 1 ha terminato il servizio al nodo 4 ed è passata al nodo 8.
```

```
L'entità 1 ha iniziato il servizio presso il server 1 del nodo 8.
```

```
L'entità 2 ha iniziato il servizio presso il server 1 del nodo 4.
```

```
CLOCK = 2.04
```

```
L'entità 3 è stata generata al nodo 4.
```

```
CLOCK = 2.77
```

```
L'entità 2 ha terminato il servizio al nodo 4 ed è passata al nodo 8.
```

```
L'entità 3 ha iniziato il servizio presso il server 1 del nodo 4.
```

```
CLOCK = 3.73
```

```
L'entità 3 ha terminato il servizio al nodo 4 ed è passata al nodo 8.
```

```
CLOCK = 5.60
```

```
L'entità 4 è stata generata al nodo 1.
```

```
L'entità 4 ha iniziato il servizio presso il server 1 del nodo 1.
```

```
CLOCK = 12.23
```

```
L'entità 5 è stata generata al nodo 1.
```

```
L'entità 5 ha iniziato il servizio presso il server 2 del nodo 1.
```

CLOCK = 12.36  
L'entità 6 è stata generata al nodo 1.

CLOCK = 12.48  
L'entità 1 ha terminato il servizio al nodo 8 ed è passata al nodo 9.  
L'entità 1 ha iniziato il servizio presso il server 1 del nodo 9.  
L'entità 2 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 12.84  
L'entità 7 è stata generata al nodo 4.  
L'entità 7 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 12.94  
L'entità 8 è stata generata al nodo 1.

CLOCK = 13.36  
L'entità 9 è stata generata al nodo 4.

CLOCK = 14.50  
L'entità 10 è stata generata al nodo 4.

CLOCK = 14.84  
L'entità 7 ha terminato il servizio al nodo 4 ed è passata al nodo 8.  
L'entità 9 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 15.21  
L'entità 9 ha terminato il servizio al nodo 4 ed è passata al nodo 8.  
L'entità 10 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 15.28  
L'entità 10 ha terminato il servizio al nodo 4 ed è passata al nodo 7.  
L'entità 10 ha iniziato il servizio presso il server 1 del nodo 7.

CLOCK = 15.96  
L'entità 11 è stata generata al nodo 4.  
L'entità 11 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 16.17  
L'entità 11 ha terminato il servizio al nodo 4 ed è passata al nodo 8.

CLOCK = 16.99  
L'entità 5 ha terminato il servizio al nodo 1 ed è passata al nodo 5.  
L'entità 5 ha iniziato il servizio presso il server 1 del nodo 5.  
L'entità 6 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 18.67  
L'entità 12 è stata generata al nodo 4.  
L'entità 12 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 19.43  
L'entità 13 è stata generata al nodo 2.  
L'entità 13 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 19.87  
L'entità 12 ha terminato il servizio al nodo 4 ed è passata al nodo 7.

CLOCK = 19.87  
L'entità 6 ha terminato il servizio al nodo 1 ed è passata al nodo 5.  
L'entità 8 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 20.10  
L'entità 13 ha terminato il servizio al nodo 2 ed è passata al nodo 7.

CLOCK = 20.55  
L'entità 1 ha terminato il servizio al nodo 9 ed è passata al nodo 13.  
L'entità 1 ha iniziato il servizio presso il server 1 del nodo 13.

CLOCK = 21.07

L'entità 14 è stata generata al nodo 3.  
L'entità 14 ha iniziato il servizio presso il server 1 del nodo 3.

CLOCK = 22.09  
L'entità 14 ha terminato il servizio al nodo 3 ed è passata al nodo 8.

CLOCK = 24.35  
L'entità 1 ha terminato il servizio al nodo 13 ed è passata al nodo 15.

CLOCK = 24.61  
L'entità 15 è stata generata al nodo 4.  
L'entità 15 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 27.19  
L'entità 15 ha terminato il servizio al nodo 4 ed è passata al nodo 7.

CLOCK = 27.76  
L'entità 2 ha terminato il servizio al nodo 8 ed è passata al nodo 11.  
L'entità 2 ha iniziato il servizio presso il server 1 del nodo 11.  
L'entità 3 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 30.32  
L'entità 2 ha terminato il servizio al nodo 11 ed è passata al nodo 14.  
L'entità 2 ha iniziato il servizio presso il server 1 del nodo 14.

CLOCK = 32.09  
L'entità 16 è stata generata al nodo 4.  
L'entità 16 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 32.30  
L'entità 16 ha terminato il servizio al nodo 4 ed è passata al nodo 5.

CLOCK = 36.18  
L'entità 5 ha terminato il servizio al nodo 5 ed è passata al nodo 12.  
L'entità 5 ha iniziato il servizio presso il server 1 del nodo 12.  
L'entità 6 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 43.76  
L'entità 2 ha terminato il servizio al nodo 14 ed è passata al nodo 15.

CLOCK = 44.44  
L'entità 17 è stata generata al nodo 4.  
L'entità 17 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 44.66  
L'entità 17 ha terminato il servizio al nodo 4 ed è passata al nodo 5.

CLOCK = 45.63  
L'entità 3 ha terminato il servizio al nodo 8 ed è passata al nodo 12.  
L'entità 7 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 47.43  
L'entità 18 è stata generata al nodo 4.  
L'entità 18 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 48.79  
L'entità 19 è stata generata al nodo 4.

CLOCK = 49.15  
L'entità 18 ha terminato il servizio al nodo 4 ed è passata al nodo 7.  
L'entità 19 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 49.31  
L'entità 19 ha terminato il servizio al nodo 4 ed è passata al nodo 7.

CLOCK = 51.06  
L'entità 8 ha terminato il servizio al nodo 1 ed è passata al nodo 7.

CLOCK = 51.08  
L'entità 6 ha terminato il servizio al nodo 5 ed è passata al nodo 10.  
L'entità 6 ha iniziato il servizio presso il server 1 del nodo 10.  
L'entità 16 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 54.35  
L'entità 20 è stata generata al nodo 3.  
L'entità 20 ha iniziato il servizio presso il server 1 del nodo 3.

CLOCK = 54.81  
L'entità 21 è stata generata al nodo 2.  
L'entità 21 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 54.91  
L'entità 7 ha terminato il servizio al nodo 8 ed è passata al nodo 10.  
L'entità 9 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 56.30  
L'entità 21 ha terminato il servizio al nodo 2 ed è passata al nodo 8.

CLOCK = 56.56  
L'entità 22 è stata generata al nodo 1.  
L'entità 22 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 60.05  
L'entità 6 ha terminato il servizio al nodo 10 ed è passata al nodo 14.  
L'entità 6 ha iniziato il servizio presso il server 1 del nodo 14.  
L'entità 7 ha iniziato il servizio presso il server 1 del nodo 10.

CLOCK = 60.70  
L'entità 23 è stata generata al nodo 4.  
L'entità 23 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 61.77  
L'entità 9 ha terminato il servizio al nodo 8 ed è passata al nodo 10.  
L'entità 11 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 62.23  
L'entità 24 è stata generata al nodo 4.

CLOCK = 62.47  
L'entità 10 ha terminato il servizio al nodo 7 ed è passata al nodo 11.  
L'entità 10 ha iniziato il servizio presso il server 1 del nodo 11.  
L'entità 8 ha iniziato il servizio presso il server 1 del nodo 7.

CLOCK = 63.05  
L'entità 23 ha terminato il servizio al nodo 4 ed è passata al nodo 8.  
L'entità 24 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 63.33  
L'entità 24 ha terminato il servizio al nodo 4 ed è passata al nodo 8.

CLOCK = 63.49  
L'entità 16 ha terminato il servizio al nodo 5 ed è passata al nodo 9.  
L'entità 16 ha iniziato il servizio presso il server 1 del nodo 9.  
L'entità 17 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 64.81  
L'entità 25 è stata generata al nodo 4.  
L'entità 25 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 64.97  
L'entità 20 ha terminato il servizio al nodo 3 ed è passata al nodo 5.

CLOCK = 65.47  
L'entità 25 ha terminato il servizio al nodo 4 ed è passata al nodo 7.

CLOCK = 65.93

L'entità 5 ha terminato il servizio al nodo 12 ed è passata al nodo 14.  
L'entità 3 ha iniziato il servizio presso il server 1 del nodo 12.

CLOCK = 67.96

L'entità 26 è stata generata al nodo 2.  
L'entità 26 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 68.04

L'entità 4 ha terminato il servizio al nodo 1 ed è passata al nodo 7.

CLOCK = 68.46

L'entità 27 è stata generata al nodo 1.  
L'entità 27 ha iniziato il servizio presso il server 1 del nodo 1.

CLOCK = 68.65

L'entità 10 ha terminato il servizio al nodo 11 ed è passata al nodo 13.  
L'entità 10 ha iniziato il servizio presso il server 1 del nodo 13.

CLOCK = 69.39

L'entità 11 ha terminato il servizio al nodo 8 ed è passata al nodo 10.  
L'entità 14 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 69.49

L'entità 6 ha terminato il servizio al nodo 14 ed è passata al nodo 15.  
L'entità 5 ha iniziato il servizio presso il server 1 del nodo 14.

CLOCK = 69.71

L'entità 26 ha terminato il servizio al nodo 2 ed è passata al nodo 6.  
L'entità 26 ha iniziato il servizio presso il server 1 del nodo 6.

CLOCK = 72.23

L'entità 28 è stata generata al nodo 4.  
L'entità 28 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 72.29

L'entità 28 ha terminato il servizio al nodo 4 ed è passata al nodo 8.

CLOCK = 72.42

L'entità 29 è stata generata al nodo 4.  
L'entità 29 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 72.48

L'entità 29 ha terminato il servizio al nodo 4 ed è passata al nodo 8.

CLOCK = 72.54

L'entità 30 è stata generata al nodo 4.  
L'entità 30 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 72.68

L'entità 31 è stata generata al nodo 1.

CLOCK = 73.43

L'entità 30 ha terminato il servizio al nodo 4 ed è passata al nodo 7.

CLOCK = 73.66

L'entità 32 è stata generata al nodo 2.  
L'entità 32 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 74.23

L'entità 27 ha terminato il servizio al nodo 1 ed è passata al nodo 8.  
L'entità 31 ha iniziato il servizio presso il server 1 del nodo 1.

CLOCK = 74.43

L'entità 14 ha terminato il servizio al nodo 8 ed è passata al nodo 9.  
L'entità 14 ha iniziato il servizio presso il server 2 del nodo 9.  
L'entità 21 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 75.01

L'entità 10 ha terminato il servizio al nodo 13 ed è passata al nodo 15.

CLOCK = 75.14

L'entità 33 è stata generata al nodo 2.

CLOCK = 75.51

L'entità 32 ha terminato il servizio al nodo 2 ed è passata al nodo 6.

L'entità 33 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 75.91

L'entità 17 ha terminato il servizio al nodo 5 ed è passata al nodo 12.

L'entità 20 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 76.08

L'entità 3 ha terminato il servizio al nodo 12 ed è passata al nodo 13.

L'entità 3 ha iniziato il servizio presso il server 1 del nodo 13.

L'entità 17 ha iniziato il servizio presso il server 1 del nodo 12.

CLOCK = 76.08

L'entità 34 è stata generata al nodo 4.

L'entità 34 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 76.63

L'entità 16 ha terminato il servizio al nodo 9 ed è passata al nodo 13.

CLOCK = 77.51

L'entità 35 è stata generata al nodo 4.

CLOCK = 77.63

L'entità 34 ha terminato il servizio al nodo 4 ed è passata al nodo 7.

L'entità 35 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 78.06

L'entità 7 ha terminato il servizio al nodo 10 ed è passata al nodo 14.

L'entità 9 ha iniziato il servizio presso il server 1 del nodo 10.

CLOCK = 79.10

L'entità 36 è stata generata al nodo 2.

CLOCK = 79.33

L'entità 33 ha terminato il servizio al nodo 2 ed è passata al nodo 7.

L'entità 36 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 80.80

L'entità 36 ha terminato il servizio al nodo 2 ed è passata al nodo 5.

CLOCK = 80.98

L'entità 37 è stata generata al nodo 1.

CLOCK = 83.87

L'entità 17 ha terminato il servizio al nodo 12 ed è passata al nodo 13.

CLOCK = 84.06

L'entità 21 ha terminato il servizio al nodo 8 ed è passata al nodo 9.

L'entità 21 ha iniziato il servizio presso il server 1 del nodo 9.

L'entità 23 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 84.37

L'entità 35 ha terminato il servizio al nodo 4 ed è passata al nodo 5.

CLOCK = 85.87

L'entità 3 ha terminato il servizio al nodo 13 ed è passata al nodo 15.

L'entità 16 ha iniziato il servizio presso il server 1 del nodo 13.

CLOCK = 87.77

L'entità 8 ha terminato il servizio al nodo 7 ed è passata al nodo 12.

L'entità 8 ha iniziato il servizio presso il server 1 del nodo 12.

L'entità 33 ha iniziato il servizio presso il server 1 del nodo 7.



CLOCK = 88.21  
L'entità 38 è stata generata al nodo 1.

CLOCK = 88.24  
L'entità 39 è stata generata al nodo 3.  
L'entità 39 ha iniziato il servizio presso il server 1 del nodo 3.

CLOCK = 88.48  
L'entità 8 ha terminato il servizio al nodo 12 ed è passata al nodo 13.

CLOCK = 88.72  
L'entità 33 ha terminato il servizio al nodo 7 ed è passata al nodo 10.  
L'entità 34 ha iniziato il servizio presso il server 1 del nodo 7.

CLOCK = 89.39  
L'entità 40 è stata generata al nodo 3.

CLOCK = 89.82  
L'entità 41 è stata generata al nodo 2.  
L'entità 41 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 90.38  
L'entità 41 ha terminato il servizio al nodo 2 ed è passata al nodo 7.

CLOCK = 91.65  
L'entità 42 è stata generata al nodo 4.  
L'entità 42 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 92.09  
L'entità 42 ha terminato il servizio al nodo 4 ed è passata al nodo 7.

CLOCK = 92.42  
L'entità 9 ha terminato il servizio al nodo 10 ed è passata al nodo 14.  
L'entità 11 ha iniziato il servizio presso il server 1 del nodo 10.

CLOCK = 92.72  
L'entità 22 ha terminato il servizio al nodo 1 ed è passata al nodo 7.  
L'entità 37 ha iniziato il servizio presso il server 2 del nodo 1.

CLOCK = 92.97  
L'entità 21 ha terminato il servizio al nodo 9 ed è passata al nodo 14.

CLOCK = 94.65  
L'entità 26 ha terminato il servizio al nodo 6 ed è passata al nodo 9.  
L'entità 26 ha iniziato il servizio presso il server 1 del nodo 9.  
L'entità 32 ha iniziato il servizio presso il server 1 del nodo 6.

CLOCK = 95.21  
L'entità 20 ha terminato il servizio al nodo 5 ed è passata al nodo 11.  
L'entità 20 ha iniziato il servizio presso il server 1 del nodo 11.  
L'entità 36 ha iniziato il servizio presso il server 1 del nodo 5.

CLOCK = 95.22  
L'entità 43 è stata generata al nodo 4.  
L'entità 43 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 95.81  
L'entità 43 ha terminato il servizio al nodo 4 ed è passata al nodo 6.

CLOCK = 96.07  
L'entità 44 è stata generata al nodo 3.

CLOCK = 97.26  
L'entità 45 è stata generata al nodo 4.  
L'entità 45 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 97.58

L'entità 23 ha terminato il servizio al nodo 8 ed è passata al nodo 12.  
L'entità 23 ha iniziato il servizio presso il server 1 del nodo 12.  
L'entità 24 ha iniziato il servizio presso il server 1 del nodo 8.

CLOCK = 98.07

L'entità 46 è stata generata al nodo 1.

CLOCK = 98.51

L'entità 45 ha terminato il servizio al nodo 4 ed è passata al nodo 8.

CLOCK = 99.30

L'entità 47 è stata generata al nodo 4.

L'entità 47 ha iniziato il servizio presso il server 1 del nodo 4.

CLOCK = 99.42

L'entità 47 ha terminato il servizio al nodo 4 ed è passata al nodo 6.

CLOCK = 99.92

L'entità 48 è stata generata al nodo 2.

L'entità 48 ha iniziato il servizio presso il server 1 del nodo 2.

CLOCK = 99.96

L'entità 16 ha terminato il servizio al nodo 13 ed è passata al nodo 15.

L'entità 17 ha iniziato il servizio presso il server 1 del nodo 13.

CLOCK = 100.70

L'entità 14 ha terminato il servizio al nodo 9 ed è passata al nodo 14.

FINE SIMULAZIONE

Avarage time into the network per entity = 55.99

Avarage time spent in the queues per entity:

Node 1: 21.11  
Node 2: 0.07  
Node 3: 37.09  
Node 4: 0.18  
Node 5: 23.38  
Node 6: 53.59  
Node 7: 45.25  
Node 8: 42.00  
Node 9: 0.00  
Node 10: 26.64  
Node 11: 0.00  
Node 12: 4.09  
Node 13: 18.97  
Node 14: 52.53  
Node 15: SINK

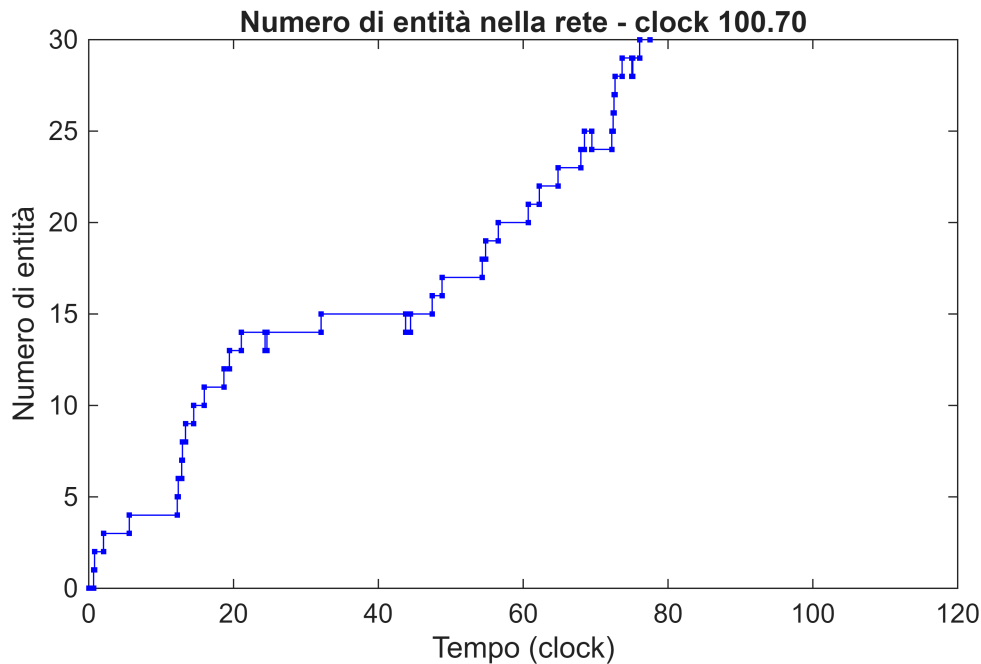
Avarage number of entities in the network = 19.79

Avarage number of entities in the nodes of the network:

Node 1: 0.40  
Node 2: 0.01  
Node 3: 0.16  
Node 4: 0.04  
Node 5: 0.95  
Node 6: 0.25  
Node 7: 4.87  
Node 8: 3.53  
Node 9: 0.00  
Node 10: 0.56  
Node 11: 0.00  
Node 12: 0.20  
Node 13: 0.37

Node 14: 0.42  
Node 15: SINK

```
xlim([0.0 120.0])  
ylim([0.0 30.0])
```



## ESEMPIO PATOLOGICO

All'interno della classe `Network` è presente un controllo che verifica l'assenza di loop nella rete. Tuttavia, questo controllo **non tiene conto dei pesi sugli archi rappresentati da function handle**. Di conseguenza, in presenza di archi condizionati da proprietà dinamiche delle entità (come `@(ent) ent.prop == 1`), il controllo non può garantire che la rete sia effettivamente priva di loop in ogni possibile configurazione. Vediamone un esempio applicativo.

```
matrice_adiacenza = {  
    0, @(ent) ent.prop == 0, 0, @(ent) ent.prop == 1;  
    0, 0, @(ent) ent.prop == 0, 0;  
    0, @(ent) ent.prop == 0, 0, @(ent) ent.prop == 1;  
    0, 0, 0, 1;  
};  
  
n = size(matrice_adiacenza, 1);  
  
% Una distribuzione di interarrivo per ciascun nodo: il -1 è un valore  
% usato per dire che il corrispondente nodo non supporta arrivi esterni.  
distr_InterArrivi = cell(n,1);  
distr_InterArrivi{1} = Exponential(20);  
distr_InterArrivi{2} = -1;  
distr_InterArrivi{3} = -1;  
distr_InterArrivi{4} = -1;  
  
% Un cell array per ciascun nodo con distribuzioni di servizio per ciascun  
% server del nodo.
```

```
distr_TempiServizio = cell(n,1);
distr_TempiServizio{1} = {Exponential(10)};
distr_TempiServizio{2} = {Exponential(10)};
distr_TempiServizio{3} = {Exponential(10)};
distr_TempiServizio{4} = {Exponential(10)};
```

*% Una cell array per definire la policy della coda di ciascun nodo.*

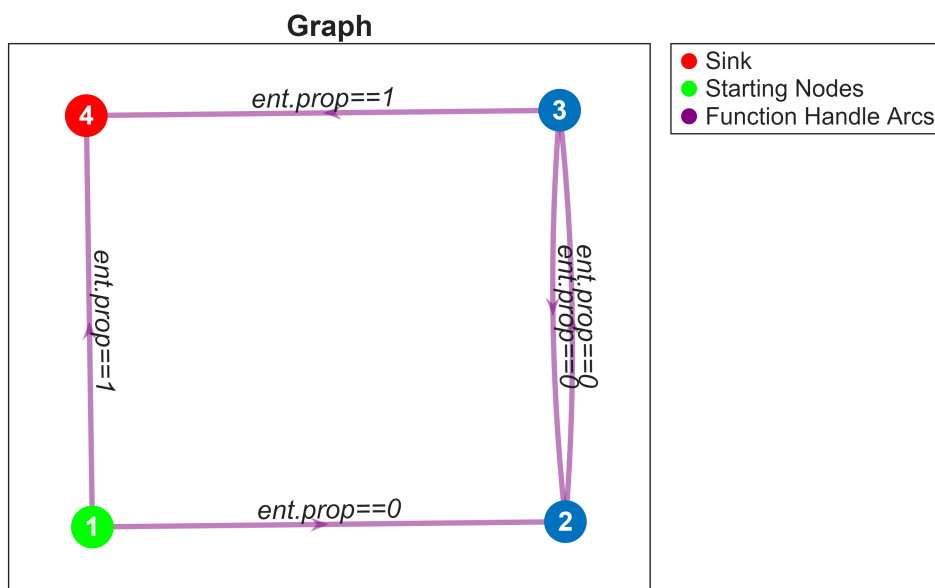
```
policy = cell(n,1);
policy{1} = 'FIFO';
policy{2} = 'FIFO';
policy{3} = 'FIFO';
policy{4} = 'FIFO';
```

*% Costruzione dell'oggetto Network*

```
net = Network(matrice_adiacenza, distr_InterArrivi, distr_TempiServizio, policy);
```

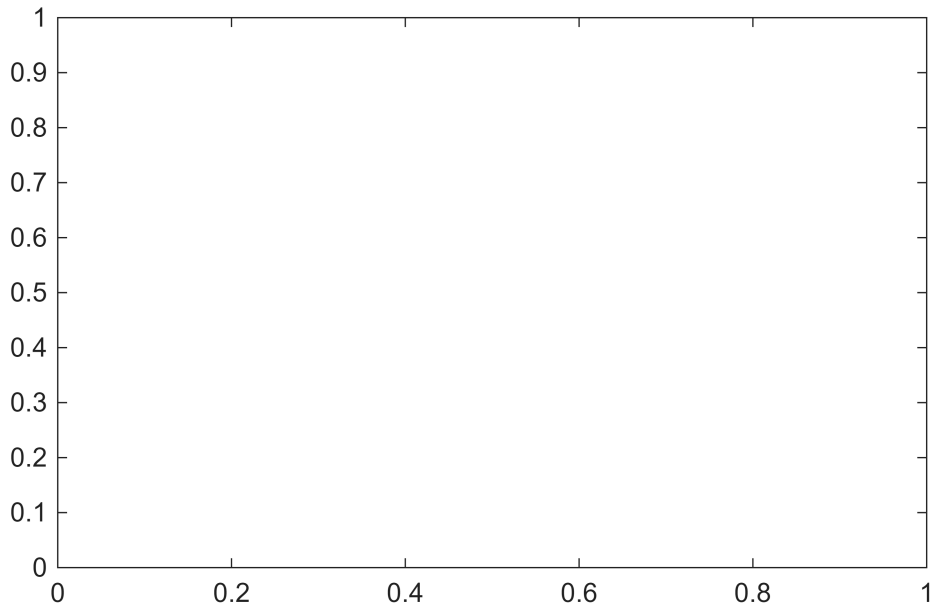
Come si può osservare dal grafo sottostante, questa rete risulta superare il test di raggiungibilità definito nella classe **Network**. Tuttavia, nella realtà, gli utenti con `prop == 0` finiscono intrappolati in un ciclo infinito all'interno del grafo.

```
stampa_grafo(net,true)
```



*% Inizializzo una simulazione*

```
attributi_entita = [];
attributi_entita.prop = {0};
sim = Simulation(net, attributi_entita);
le = sim.run(100,true);
```



```
*****
INIZIO SIMULAZIONE ...
CLOCK = 63.15
Matrice del Percorso:
  0   1   0   0
  0   0   1   0
  0   1   0   0
  0   0   0   1
Error using Entity (line 102)
Non è possibile raggiungere nessuno dei nodi di arrivo partendo dal nodo di partenza.

Error in ArrivalEvent/process (line 21)
    ent = Entity(clock, sim.attributi_entita, self.nodoID, sim);
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Error in Simulation/run (line 84)
    evento_da_processare.process(self.clock, self);
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Come possiamo osservare, abbiamo creato un caso patologico in cui il percorso degli utenti con proprietà `== 0` si traduce in un loop infinito, impedendo loro di uscire dalla rete. Questa problematica si manifesta solo una volta avviata la simulazione, e non durante la fase di creazione della rete.

## Applicazione All'Ottimizzazione

La libreria proposta consente la simulazione di sistemi modellizzabili tramite reti di Jackson, come ad esempio alcuni semplici sistemi logistici di produzione, tra cui i *job shop* e i *flow shop*. Questo permette di valutare e confrontare diverse configurazioni di parametri, quali il numero di server o le politiche di gestione delle code, al fine di individuare soluzioni più efficienti e ottimizzare le principali metriche di performance, come il flow time medio.

## Limitazioni e Possibili Estensioni

La libreria proposta è costituita da alcune classi base, e pertanto non copre in modo completo tutti i possibili sistemi a code. La limitazione principale è che essa è progettata per simulare esclusivamente sistemi modellizzabili tramite reti di Jackson.

Sono inoltre presenti altre limitazioni che, pur non attualmente supportate, potrebbero essere affrontate in future estensioni. Tra queste, ad esempio, vi è l'assenza del supporto per code a capacità finita, che rappresenta una funzionalità utile in contesti reali come la gestione di risorse fisicamente limitate.