The Offered-Load in Fork-Join Networks
Applications to Staffing of Emergency Departments

Itamar Zaied
Jointly with H.Kaspi and A.Mandelbaum

May 25, 2011
The Offered Load: The Stationary Case

hours of work (= service) that arrive per hour.

Example:
\( \lambda = 20 \) patients/hour; \( E[S] = 0.5 \) hours. Offered-Load \( R = 20 \cdot 0.5 = 10 \) hour of work per hour.

The Offered Load of an \( M_t/GI/N_t \) queue

For the \( M_t/GI/N_t \) queue, the offered load \( R = \{ R(t), t \geq 0 \} \) is given by the function \( R(t) = E[L(t)] \), where \( L(t) \) is the number of customers/patients (=number of busy servers) at time \( t \), in the corresponding \( M_t/\infty \) queue.
The Offered Load: The Stationary Case

hours of work (= service) that arrive per hour.

Example:
\( \lambda = 20 \) patients/hour; \( E[S] = 0.5 \) hours. Offered-Load \( R = 20 \cdot 0.5 = 10 \) hour of work per hour.

The Offered Load of an \( M_t/GI/N_t \) queue

For the \( M_t/GI/N_t \) queue, the offered load \( R = \{ R(t), t \geq 0 \} \) is given by the function \( R(t) = E[L(t)] \), where \( L(t) \) is the number of customers/patients (=number of busy servers) at time \( t \), in the corresponding \( M_t/GI/\infty \) queue.

- Offered-Load has been proved to be the skeleton for staffing of time-varying systems.
A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.
A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.
A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.
Fork-Join Networks

A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.
A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.
A fork-join network consists of service stations, which serve arriving customers simultaneously and sequentially according to pre-designed deterministic precedence constraints.
In his MSc thesis, Yariv Marmur developed a generic ED simulation, using Rockwell’s software ”Arena”.

**Diagram:**

- The diagram shows a simulation model for an ED (Emergency Department) with various patient categories and hospital rooms.
- The model includes rooms labeled as DR, Nurse, Room1, etc., indicating different departments or areas within the ED.
- There are icons representing patients and their status, such as X-Ray, CT, Lab, Consult, Hold.
- The layout includes labels for X-Ray, CT, Lab, and HELEM, which are likely different types of patient care or departments within the ED.
Offered-Load calculation using the ED simulator:
Offered-Load calculation using the ED simulator:

- In analytical models, the Offered-Load is the number of patients in the corresponding infinite-server network.
Offered-Load calculation using the ED simulator:

- In analytical models, the Offered-Load is the number of patients in the corresponding infinite-server network.
- We thus calculate the Offered-Load using the ED simulator, where we set the number of resources to be $\infty$ (at all the resources or only the resources under consideration).
The Offered-load of Resources: Doctors, Nurses

Internal Doctors Offered load

Internal Nurses Offered load

Surgical Doctors Offered load

Surgical Nurses Offered load

Orthopedic Doctors Offered load
Zoom in on Internal Doctors

Internal Doctors Offered load

Number of busy agents

Time

DrInternal_1WeekDay
DrInternal_1WeekEnd
DrInternal_4WeekDay
DrInternal_4WeekEnd
Using the square-root staffing rule:

\[ N(t) = \text{round}(R(t) + \beta \cdot \sqrt{R(t)}) , \]

where:

\[ \text{round}(x) = \begin{cases} 
\lceil x \rceil & \text{if } \lceil x \rceil - x \geq 0.5 \\
\lfloor x \rfloor & \text{if } \lceil x \rceil - x < 0.5 
\end{cases} \]
Using the square-root staffing rule:

\[ N(t) = \text{round}(R(t) + \beta \cdot \sqrt{R(t)}) , \]

where:

- \( N(t) \) are the number of agents at time \( t \).
Using the square-root staffing rule:

\[ N(t) = \text{round}(R(t) + \beta \cdot \sqrt{R(t)}) \],

where:

- \( N(t) \) are the number of agents at time \( t \).
- \( \beta \) is "safety staffing".
Using the square-root staffing rule:

\[ N(t) = \text{round}(R(t) + \beta \cdot \sqrt{R(t)}) \]

where:

- \( N(t) \) are the number of agents at time \( t \).
- \( \beta \) is "safety staffing".
- \( \text{round}(x) = \begin{cases} \left\lceil x \right\rceil & \text{if } \left\lceil x \right\rceil - x \geq 0.5 \\ \left\lfloor x \right\rfloor & \text{if } \left\lceil x \right\rceil - x < 0.5 \end{cases} \)
Length of Stay

- Rambam Schedule
- beta=0.7
- Beta=1.5

0:00 1:00 2:00 3:00 4:00 5:00 6:00 7:00 8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00

0 50 100 150 200 250 300 350
Time-Stable Measure of Performance

Time Up to First Physician Visit

Time Before First Physician Visit

- Rambam Schedule
- Beta=0.7
- Beta=1.5
Different $\beta$’s vs Original Staffing

Anatomy of Sojourn Time

Current Staffing
LOS=7.1 hours

- Queue Time: 8%
- Service Time: 23%
- Clinical Treatment Time: 12%
- Sync Time: 57%

$\beta = 0.7$
LOS=6 hours

- Queue Time: 12%
- Service Time: 26%
- Clinical Treatment Time: 15%
- Sync Time: 47%

$\beta = 1.5$
LOS=4.6 hours

- Queue Time: 17%
- Service Time: 35%
- Clinical Treatment Time: 20%
- Sync Time: 28%

Daily Hours of Internal Dr: 56

Daily Hours of Internal Nurses: 48

Daily Hours of Internal Dr: 63

Daily Hours of Internal Nurses: 41

Daily Hours of Internal Dr: 86

Daily Hours of Internal Nurses: 68
The Black Box Method

Consider a fork join network with $V$ its set of nodes and $A$ its set of arcs. To calculate the Offered Load of station $i$, one can define a new fork join network $i^-$, as follows:
Let $V^-$ be all the nodes of $V$ which have a directed path to station $i$. 

Example: the network $10^-$
The Black Box Method

- Let $V^-$ be all the nodes of $V$ which have a directed path to station $i$.
- Let $A^-$ be all the arcs in $A$ which connect a node from $V^-$ to a node from $V^- \cup i$.

Example: the network $10^-$
The Black Box Method

- Let \( V^- \) be all the nodes of \( V \) which have a directed path to station \( i \).
- Let \( A^- \) be all the arcs in \( A \) which connect a node from \( V^- \) to a node from \( V^- \cup i \).
- Let \( i^- \) be the fork-join network with \( V^- \) its set of nodes and \( A^- \) its set of arcs.

Example: the network 10^-
We now have a Tandem of 2 service stations, where queue $i$ is the second queue.

The offered load at station $i$, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where $T_i$ is the total sojourn time of network $i^-$ and $S_i^e$ is the residual service time at station $i$. 

The Offered-Load at station $i$

The offered load at station $i$, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where $T_i$ is the total sojourn time of network $i^-$ and $S_i^e$ is the residual service time at station $i$. 

Itamar Zaied Jointly with H.Kaspi and A.Mandelbaum (2011)
We now have a Tandem of 2 service stations, where queue $i$ is the second queue. The offered load at station $i$, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where $T_i$ is the total sojourn time of network $i^-$ and $S_i^e$ is the residual service time at station $i$.

- Thus, finding the distribution of $T_i$ is required for calculating the Offered-Load of station $i$. 

---

**The Offered-Load at station $i$**

The offered load at station $i$, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where $T_i$ is the total sojourn time of network $i^-$ and $S_i^e$ is the residual service time at station $i$.
We now have a Tandem of 2 service stations, where queue $i$ is the second queue.

The offered load at station $i$, is given by

$$R_i(t) = E(\lambda(t - T_i - S_i^e)) \cdot E(S_i), \quad t \geq 0,$$

where $T_i$ is the total sojourn time of network $i^-$ and $S_i^e$ is the residual service time at station $i$.

- Thus, finding the distribution of $T_i$ is required for calculating the Offered-Load of station $i$.
- $S_i^e$’s density function is $f_{S_i^e}(x) = \frac{1 - F_{S_i}(x)}{E(S_i)}$. 

The Offered-Load at station $i$
The periodic nature of arrivals allows one to calculate the Offered-Load in an easier way: Discrete Fourier transform. The example of calculations were applied to the following fork-join network.
The periodic nature of arrivals allows one to calculate the Offered-Load in an easier way: **Discrete Fourier transform.**
The periodic nature of arrivals allows one to calculate the Offered-Load in an easier way: Discrete Fourier transform.

The example of calculations were applied to the following fork-join network:
Periodic Arrival Rate: Actual Arrival Rate

Alternative Operation - C
Recourse Queue - Synchronization Queue -
Ending point of alternative operation -
Periodic Arrival Rate: Actual Arrival Rate

Example: The Offered-Load of Internal Drs:

1. We look at the activities that are executed by Internal Dr’s: ”First Examination”, ”Treatment” and ”Decision”.

Itamar Zaied Jointly with H.Kaspi and A.Mandelbaum

The Offered-Load in Fork-Join Networks

May 25, 2011 23 / 35
Example: The Offered-Load of Internal Drs:

1. We look at the activities that are executed by Internal Dr’s: ”First Examination”, ”Treatment” and ”Decision”.
2. We then calculate the Offered-Load of each one of the activities.
Example: The Offered-Load of Internal Drs:

1. We look at the activities that are executed by Internal Dr’s: ”First Examination”, ”Treatment” and ”Decision”.
2. We then calculate the Offered-Load of each one of the activities.
3. Summing up the Offered-Load of the activities will give us the Offered-Load of Internal Dr’s.
Example: The Offered-Load of Internal Drs:

\[ R(t) = \sum_{j=1}^{3} R_{aj}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i k N}{N}} E(e^{-\frac{2\pi i k N}{N} k S}) \cdot E(S_{aj}), \]

where \( a_1 = "First\ Examination" \), \( a_2 = "Treatment" \) and \( a_3 = "Decision" \).
Example: The Offered-Load of Internal Drs:

\[ R(t) = \sum_{j=1}^{3} R_{aj}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{aj}), \]

where \( a_1 = "First\ Examination" \), \( a_2 = "Treatment" \) and \( a_3 = "Decision" \).

\( x_0, ..., x_{N-1} \) are the arrival rates at times \( t_0 = 0, ... t_{N-1} = N - 1 \).
Example: The Offered-Load of Internal Drs:

\[
R(t) = \sum_{j=1}^{3} R_{aj}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E\left(e^{-\frac{2\pi i}{N} kS}\right) \cdot E(S_{aj}),
\]

where \( a_1 = "\text{First Examination}" \), \( a_2 = "\text{Treatment}" \) and \( a_3 = "\text{Decision}" \).

- \( x_0, ..., x_{N-1} \) are the arrival rates at times \( t_0 = 0, ... t_{N-1} = N - 1 \).
- \( X_k \) is the discrete Fourier transform of \( x_0, ..., x_{N-1} \)
  \[
  (X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, ..., N - 1),
  \]
- \( n = \{ k \mod(N) | k = \max\{m : m \leq t\} \} \).
Example: The Offered-Load of Internal Drs:

\[ R(t) = \sum_{j=1}^{3} R_{aj}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{aj}), \]

where \( a_1 = "First Examination" \), \( a_2 = "Treatment" \) and \( a_3 = "Decision" \).

- \( x_0, \ldots, x_{N-1} \) are the arrival rates at times \( t_0 = 0, \ldots t_{N-1} = N - 1 \).
- \( X_k \) is the discrete Fourier transform of \( x_0, \ldots, x_{N-1} \)
  \( (X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \ldots, N - 1) \),
- \( n = \{ k \mod(N) \mid k = \max\{ m : m \leq t \} \} \).
- \( S_a \) is the service time of activity \( a \).
Example: The Offered-Load of Internal Drs:

\[ R(t) = \sum_{j=1}^{3} R_a_j(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i \cdot kn}{N}} E(e^{-\frac{2\pi i \cdot kS}{N}}) \cdot E(S_{a_j}), \]

where \( a_1 = "First\ Examination", a_2 = "Treatment" \) and \( a_3 = "Decision". \)

- \( x_0, ..., x_{N-1} \) are the arrival rates at times \( t_0 = 0, ... t_{N-1} = N - 1 \).
- \( X_k \) is the discrete Fourier transform of \( x_0, ..., x_{N-1} \) \( (X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i \cdot kn}{N}} \quad k = 0, ..., N - 1), \)
  - \( n = \{k \mod(N) | k = \max\{m : m \leq t\}\}. \)
- \( S_a \) is the service time of activity \( a \).
- \( S \) is the completion time of the network that starts at the ED entry and ends at activity \( a \).
Example: The Offered-Load of Internal Drs:

\[ R(t) = \sum_{j=1}^{3} R_{aj}(t) = \sum_{j=1}^{3} \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} E(e^{-\frac{2\pi i}{N} kS}) \cdot E(S_{aj}), \]

where \( a_1 = "\text{First Examination}" \), \( a_2 = "\text{Treatment}" \) and \( a_3 = "\text{Decision}" \).

- \( x_0, \ldots, x_{N-1} \) are the arrival rates at times \( t_0 = 0, \ldots t_{N-1} = N - 1 \).
- \( X_k \) is the discrete Fourier transform of \( x_0, \ldots, x_{N-1} \)\n  \( (X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \ldots, N - 1) \),
- \( n = \{ k \mod(N)|k = \max\{m : m \leq t\} \} \).
- \( S_a \) is the service time of activity \( a \).
- \( S \) is the completion time of the network that starts at the ED entry and ends at activity \( a \).
- To calculate the expression \( E(e^{-\frac{2\pi i}{N} kS}) \) we disaggregated the complex ED network into a few simpler subnetworks.
Periodic Arrival Rate: Actual Arrival Rate
Comparison with the Offered-Load, calculated using the simulator

- The method provides an Offered-Load approximation. Its advantages are:
Comparison with the Offered-Load, calculated using the simulator

- The method provides an Offered-Load approximation. Its advantages are:
  - Easy to calculate, once a calculation algorithm has been established.
Comparison with the Offered-Load, calculated using the simulator

- The method provides an Offered-Load approximation. Its advantages are:
  1. Easy to calculate, once a calculation algorithm has been established.
  2. One can use this method on other fork-join networks (other ED networks in particular); instead of programming a simulator.
Comparing results: Simulator vs Approximation

For example, results of internal Drs (staffing with $\beta = 0.7$):
We adopt a result by Adlakha and Kulkarni [1986], to represent a fork-join network as a continues time Markov chain (activities on arcs vs activities on nodes, in our case).
We adopt a result by Adlakha and Kulkarni [1986], to represent a fork-join network as a continues time Markov chain (activities on arcs vs activities on nodes, in our case).

The continues time Markov chain state space consists of all the pairs $(B, D)$: $B$ denote all the active nodes and $D$ all the dormant nodes.
The fork-join Network: (here, $\mu_i$ is the service rate of station $i$)

The CTMC rate diagram:

- **Phase 2**
  - $B = \{1\}$
  - $D = \{3\}$
  - Transition rates: $\mu_2$

- **Phase 1**
  - $B = \{1, 2\}$
  - $D = \emptyset$
  - Transition rates: $\mu_1$

- **Phase 4**
  - $B = \{3\}$
  - $D = \emptyset$
  - Transition rates: $\mu_3$

- **Phase 5**
  - $B = \emptyset$
  - $D = \emptyset$
Project Completion Time: Example

The fork-join Network: (here, \( \mu_i \) is the service time of station \( i \))

The CTMC rate diagram:

- Completion of Activity 2
- Phase 2: \( B = \{1\}, D = \{3\} \)
- Phase 1: \( B = \{1, 2\}, D = \phi \)
- Phase 4: \( B = \{3\}, D = \phi \)
- Phase 5: \( B = \phi, D = \phi \)
The fork-join Network: (here, $\mu_i$ is the service time of station $i$)

Completion of Activity 2

Phase 2

$B = \{1\}$
$D = \{3\}$

Phase 1

$B = \{1, 2\}$
$D = \emptyset$

Phase 4

$B = \{3\}$
$D = \emptyset$

Phase 5

$B = \emptyset$
$D = \emptyset$

Activity 1 remains active
The fork-join Network: (here, $\mu_i$ is the service time of station $i$)

Completion of Activity 2

Activity 1 remains active

Activity 3 becomes dormant, i.e. it can only become active after both Activity 1 and 2 are finished.

The CTMC rate diagram:

Phase 1
$B = \{1, 2\}$
$D = \phi$
$
\mu_1 \\
\mu_2$

Phase 2
$B = \{1\}$
$D = \{3\}$

Phase 3
$B = \{1\}$
$D = \{3\}$

Phase 4
$B = \{3\}$
$D = \phi$

Phase 5
$B = \phi$
$D = \phi$
The project completion time can be calculated via the next algorithm:

Backward Algorithm

The cdf of the PERT network service time is $F(t) = p_0(t)$. Here $p_i(t) = P(X(t) = N | X(0) = i)$, $0 \leq i \leq N$, where $X(t)$, $t \geq 0$, is the state of the project at time $t$. $p_i(t)$ are given by

$$p'_i(t) = \sum_{j \leq i} q_{ij} p_j(t)$$

$p_i(0) = \delta_{iN}$, $0 \leq i \leq N$, where $\delta_{ij} = 1$ if $i = j$, and 0 otherwise, and $q_{ij}$ are taken from the infinitesimal generator matrix of the continues time Markov chain.

We start the algorithm with $p_N(t) \equiv 1$, for $t \geq 0$ and compute $p_{N-1}(t), \ldots, p_1(t), p_0(t)$ recursively backward.
The project completion time can be calculated via the next algorithm:

**Backward Algorithm**

- The cdf of the PERT network service time is \( F(t) = p_0(t) \).
The project completion time can be calculated via the next algorithm:

**Backward Algorithm**

- The cdf of the PERT network service time is $F(t) = p_0(t)$.
- Here $p_i(t) = P(X(t) = N | X(0) = i); \quad 0 \leq i \leq N$, where
The project completion time can be calculated via the next algorithm:

### Backward Algorithm

- The cdf of the PERT network service time is $F(t) = p_0(t)$.
- Here $p_i(t) = P(X(t) = N|X(0) = i)$; $0 \leq i \leq N$, where
  - The continues time Markov chain state space is numbered as \{1, 2, ..., $N$\}. 

<table>
<thead>
<tr>
<th>States</th>
<th>Transition Probability</th>
<th>Backward Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_N$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
<tr>
<td>$p_{N-1}$</td>
<td>$q_{N-1}$</td>
<td>$p_{N-1}(t) = \sum_{j \leq i} q_{ij} p_j(t) p_{i0}(0) = \delta_{ij}$</td>
</tr>
<tr>
<td>$...$</td>
<td>$...$</td>
<td>$...$</td>
</tr>
<tr>
<td>$p_0$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
</tbody>
</table>
The project completion time can be calculated via the next algorithm:

**Backward Algorithm**

- The cdf of the PERT network service time is \( F(t) = p_0(t) \).
- Here \( p_i(t) = P(X(t) = N|X(0) = i); \quad 0 \leq i \leq N \), where
  1. The continues time Markov chain state space is numbered as \( \{1, 2, ..., N\} \).
  2. \( X(t), t \geq 0 \), is the state of the project at time \( t \).
The project completion time can be calculated via the next algorithm:

**Backward Algorithm**

1. The cdf of the PERT network service time is \( F(t) = p_0(t) \).
2. Here \( p_i(t) = P(X(t) = N | X(0) = i); \ 0 \leq i \leq N \), where
   - The continues time Markov chain state space is numbered as \( \{1, 2, ..., N\} \).
   - \( X(t), t \geq 0 \), is the state of the project at time \( t \).
   - \( p_i(t) \) are given by
     \[
     p'_i(t) = \sum_{j \leq i} q_{ij} p_j(t)
     \]
     \[
     p_i(0) = \delta_{iN}, \ 0 \leq i \leq N,
     \]
     where \( \delta_{ij} = 1 \) if \( i=j \), and 0 otherwise, and \( q_{ij} \) are taken from the infinitesimal generator matrix of the continues time Markov chain.
The project completion time can be calculated via the next algorithm:

**Backward Algorithm**

- The cdf of the PERT network service time is $F(t) = p_0(t)$.
- Here $p_i(t) = P(X(t) = N | X(0) = i); \ 0 \leq i \leq N$, where
  1. The continues time Markov chain state space is numbered as \{1, 2, ..., N\}.
  2. $X(t), \ t \geq 0$, is the state of the project at time $t$.
  3. $p_i(t)$ are given by

$$p_i'(t) = \sum_{j \leq i} q_{ij} p_j(t)$$

$$p_i(0) = \delta_{iN}, \ 0 \leq i \leq N,$$

where $\delta_{ij} = 1$ if $i=j$, and 0 otherwise, and $q_{ij}$ are taken from the infinitesimal generator matrix of the continues time Markov chain.

- We start the algorithm with $p_N(t) \equiv 1$, for $t \geq 0$ and compute $p_{N-1}(t),..., p_1(t), p_0(t)$ recursively backward.
Project Completion Time by Continuous Time Markov Chain: Example

- \( P_5(t) = 1 \),
- \( P'_3(t) = \mu_2 \cdot P_4(t) \),
- \( P'_1(t) = \mu_1 \cdot P_3(t) + \mu_2 \cdot P_2(t) \).

- \( P'_4(t) = \mu_3 \cdot P_5(t) \),
- \( P'_2(t) = \mu_1 \cdot P_4(t) \).
Staffing according to the Offered Load (square root staffing) reduces patients length of stay at the ED.
Summary

- Staffing according to the Offered Load (square root staffing) reduces patients length of stay at the ED.
- Staffing according to the Offered Load stabilizes measures of performance of the emergency department over time.
Summary

- Staffing according to the Offered Load (square root staffing) reduces patients length of stay at the ED.
- Staffing according to the Offered Load stabilizes measures of performance of the emergency department over time.
- Tractable analysis of a complex network (fork-join) captures the full complexity of an emergency department.