Service Engineering

Class 4

The Second Prerequisite: Operational Models; Service (Processing, Flow, Queueing) Networks, DSPERTs

- Review: The First Prerequisite - Data, Measurement;
- **Service Networks = Queueing Networks**;
- The Service (Processing, Flow, Queueing) Network Paradigm;
- Dynamic-Stochastic PERT/CPM models, or "Why Queues?";
- **Operational Queues**: Synchronization, Scarce Resources;
- Analyzing DS-PERT/CPM's:
  1. **Can we do it?** Answer via "Capacity Analysis"
  2. **How long will it take?** via "Response-Time Analysis"
  3. **Can we do better?** "Parametric / Sensitivity (What-If) Analysis"
  4. **What is the best we (one) can do?** "Optimization"
- Multi-Project Management.

The Second Prerequisite: (Operational) Models

Empirical Models

- Conceptual
  - Service-Process **Data** = **Flow** Network
  - **Service Networks = Queueing Networks**
- Descriptive
  - QC-Tools: Pareto, Gantt, Fishbone Diagrams,...
  - Histograms, Hazard-Rates, ...
  - Data-MOCCA: Repository + Interface
- Explanatory
  - Nonparametric: Comparative Statistics, Regression,...
  - Parametric: Log-Normal Services, (Doubly) Poisson Arrivals, Exponential (Im)Patience

Analytical Models

- Fluid (Deterministic) Models
- Stochastic Models (Birth & Death, $G/G/n$, Jackson,...)

Conceptual Models: Service Networks = Queueing Networks

- **People**, waiting for service (resource): teller, repairman, ATM;
- **Telephone-calls**, to be answered: busy, music, information;
- **Forms**, to be sent, processed, printed; for a partner (synchronization);
- **Projects**, to be planned, approved, implemented;
- **Justice**, to be made: pre-trial, hearing, retrial;
- **Ships**, for a pilot, berth, unloading crew;
- **Patients**, for an ambulance, emergency room, operation;
- **Cars**, in rush-hour, for parking;
- **Passengers** at Airports, security-check, check-in, taking-off;
- **Checks**, waiting to be processed, cashed.

Operational Queues (as opposed to, say, "weather queues"), due to:

- Scarce Resources (Resource Queues)
- Synchronization Gaps (Synchronization Queues)

Queues are costly, but (many) are **here to stay**.

Conceptual Fluid Model

Customers/units are modeled by **fluid (continuous) flow**.

Labor-day Queueing at Niagara Falls

- Appropriate when **predictable variability** prevalent;
- Useful **first-order** models/approximations, often **suffice**;
- Rigorously justifiable via Functional Strong Laws of Large Numbers.

The Service (Processing) Network Paradigm

Dynamic Stochastic Networks (Time, Uncertainty, System):
Service- / Flow- / Processing- / Queueing-Networks.

Building-Blocks:

1. **Customers** (jobs) are Served, **Flow**, Processed;
   Attributes: Arrivals, Services, Routes, Patience,...
2. **Activities** (tasks, services) are what the "jobs" are made of;
   Attributes: **Partially ordered** via Precedence-Constraints, summarized in an **Activity (Precedence) Graph** (nodes = activities, arcs = precedences).
3. **Resources** serve the Customers (perform the Activities);
   Attributes: **Scarce**, limited by **Processing (Dynamic) Capacity** (maximal sustainable service rate; in discrete events, capacity also equals the reciprocal of average service-time); Customers' Constituency, Pools, ..., summarized in a **Resource-Graph** (nodes = queues + resource-pools, arcs = flows).
4. **Queues** (Buffers) are where activities (customers) wait for their service-process to continue; **Human** (vs. Inventories)
   Attributes: Storage (Static) Capacity, which could be infinity; Operational queues are either **Resource-Queues** (waiting for a resource to become available) or **Synchronization-Queues** (waiting for a precedence-constraint to be fulfilled).
5. **Protocols** embody **information** for admission, routing, scheduling, data-archival and retrieval, quality-monitoring, performance measures (definition, monitoring),...

The Service-Network Paradigm - 2

An attempt at a definition:
The **Service-System** is envisioned (modeled) as a **graph** whose nodes represent either **activities** or **resources+queues**; **customers** flow (routed) through the system as their **tasks** are being performed by the resources; tasks processing adheres to **precedence** constraints and each resource serves the tasks within its **constituency**, following the appropriate protocol.

Schematic (Conceptual) Descriptions (in Homework):

1. **Activity** Diagram (Graph)
2. **Resource** Diagram (Graph) (Resource + Synch. Q's)
3. **Combined** (Activity+Resource) Graph
4. Information Flow

Summarized as **"Service (Process) Flow"**,
for example "Patient Flow" through hospitals (Standard **LD.3.15** of the JC**AHO** = Accreditation of Healthcare Organizations).

Historical Evolution, via buzz-words:

- **TQM** = Total Quality Management (80's)
- **BPR** = Business Process ReEngineering (90's)
- **CRM** = Customers Relationship (Revenue) Management (00's)
- BI / BA = Business Information / Analytics

Personally: From Project to Process Management
(in New Product Development, Multi-Project Management)

The Service-Network Paradigm - 3

Three (sometimes Four) Steps in Analyzing a Service Networks (demonstrated in the sequel via DS-PERTs).
Gives rise to the following **Guiding Questions**:

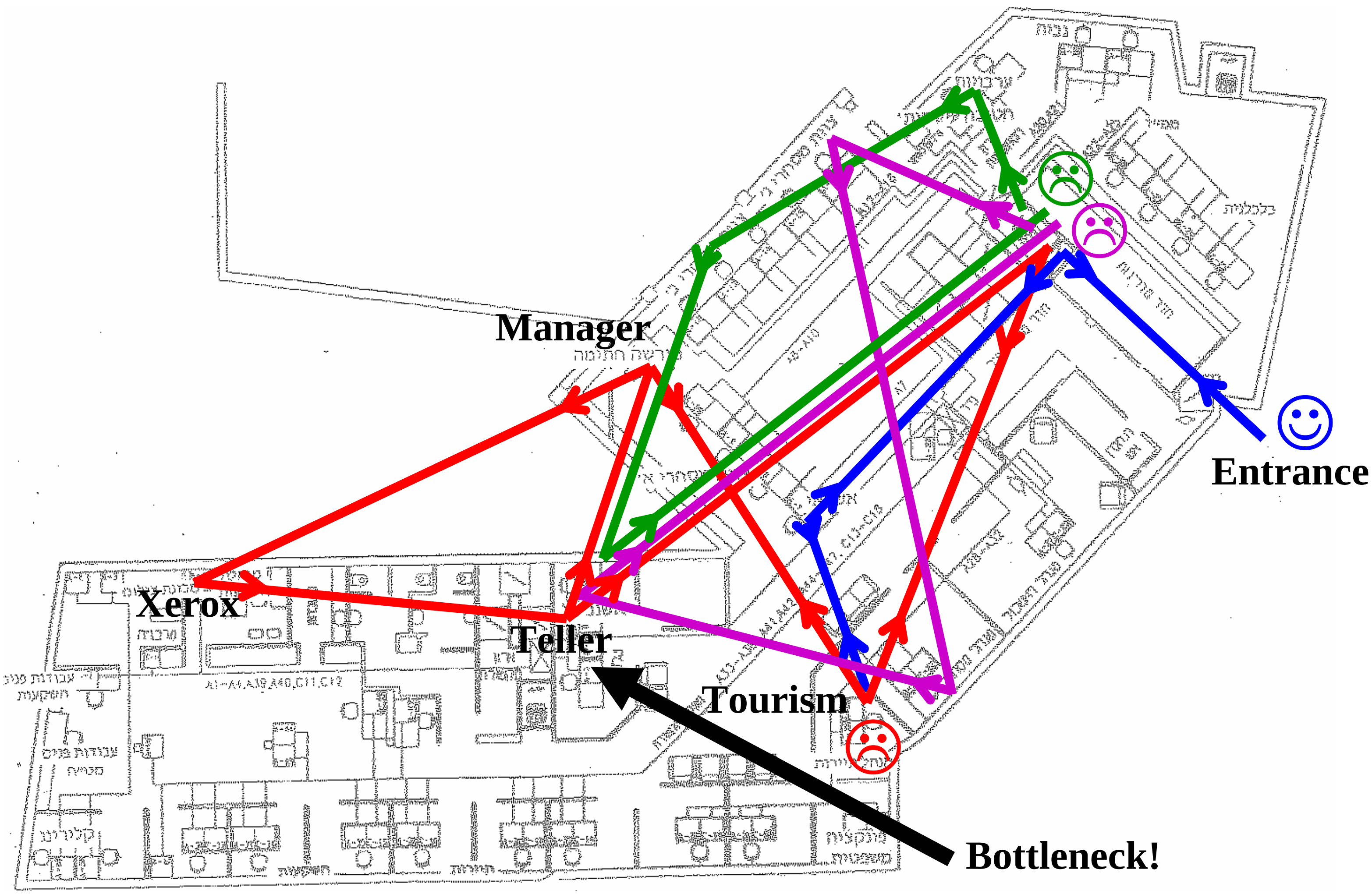
1. **Can we do it?** Deterministic **capacity analysis**, via service (process) flow diagrams (spreadsheets, linear programming), which identifies resource-bottlenecks (or at least candidates) and yields **utilization profiles**.

2. **How long will it take?** Typically stochastic **response-time analysis**, via analytical queueing-network models (exact, approximations) or simulations, which yields **congestion curves**.
   Note: When predictable variability prevails and dominates then the **Fluid View** is appropriate; the analysis is then deterministic, for example via queueing-buildup diagrams. (e.g. Recitation today, Trucks in National Cranberries next class.)

3. **Can we do better?** **Sensitivity and Parametric (what-if) analysis**, of MOPs or scenarios, which yields directions and magnitudes for **improvements**.

4. **How much better can we (one) do?** or simply: What is optimal to do? Optimal control (exact, asymptotic), typically difficult but more and more feasible, which yields optimal **protocols** (strategies, policies).

Conceptual Model:
Bank Branch = Queueing Network



Bank Branch: A Queuing Network

Transition Frequencies Between Units in The Private and Business Sections:

	To Unit / From Unit	Private Banking				Business				
		Bankers	Authorized Personal	Compens - - ations	Tellers	Tellers	Overdrafts	Authorized Personal	Full Service	Exit
Private Banking	Bankers		1%	1%	4%	4%	0%	0%	0%	90%
	Authorized Personal	12%		5%	4%	6%	0%	0%	0%	73%
	Compensations	7%	4%		18%	6%	0%	0%	1%	64%
	Tellers	6%	0%	1%		1%	0%	0%	0%	90%
Services	Tellers	1%	0%	0%	0%		1%	0%	2%	94%
	Overdrafts	2%	0%	1%	1%	19%		5%	8%	64%
	Authorized Personal	2%	1%	0%	1%	11%	5%		11%	69%
	Full Service	1%	0%	0%	0%	8%	1%	2%		88%
	Entrance	13%	0%	3%	10%	58%	2%	0%	14%	0%

Legend:

0%-5%	5%-10%	10%-15%	>15%

Dominant Paths - Business:

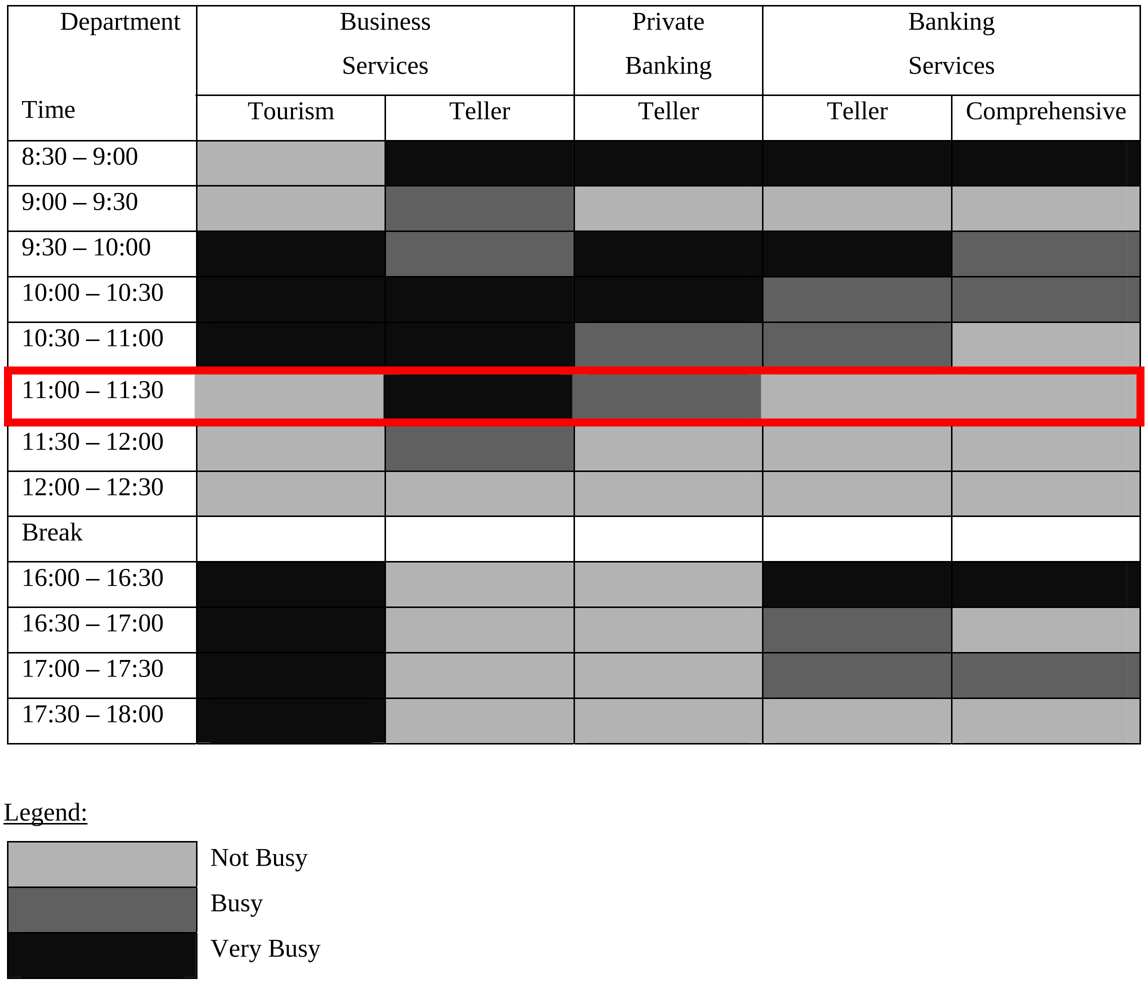
Unit / Parameter	Station 1 Tourism	Station 2 Teller	Total Dominant Path
Service Time	12.7	4.8	17.5
Waiting Time	8.2	6.9	15.1
Total Time	20.9	11.7	32.6
Service Index	**0.61**	**0.41**	**0.53**

Dominant Paths - Private:

Unit / Parameter	Station 1 Banker	Station 2 Teller	Total Dominant Path
Service Time	12.1	3.9	16.0
Waiting Time	6.5	5.7	12.2
Total Time	18.6	9.6	28.2
Service Index	**0.65**	**0.40**	**0.56**

Service Index = % time being served

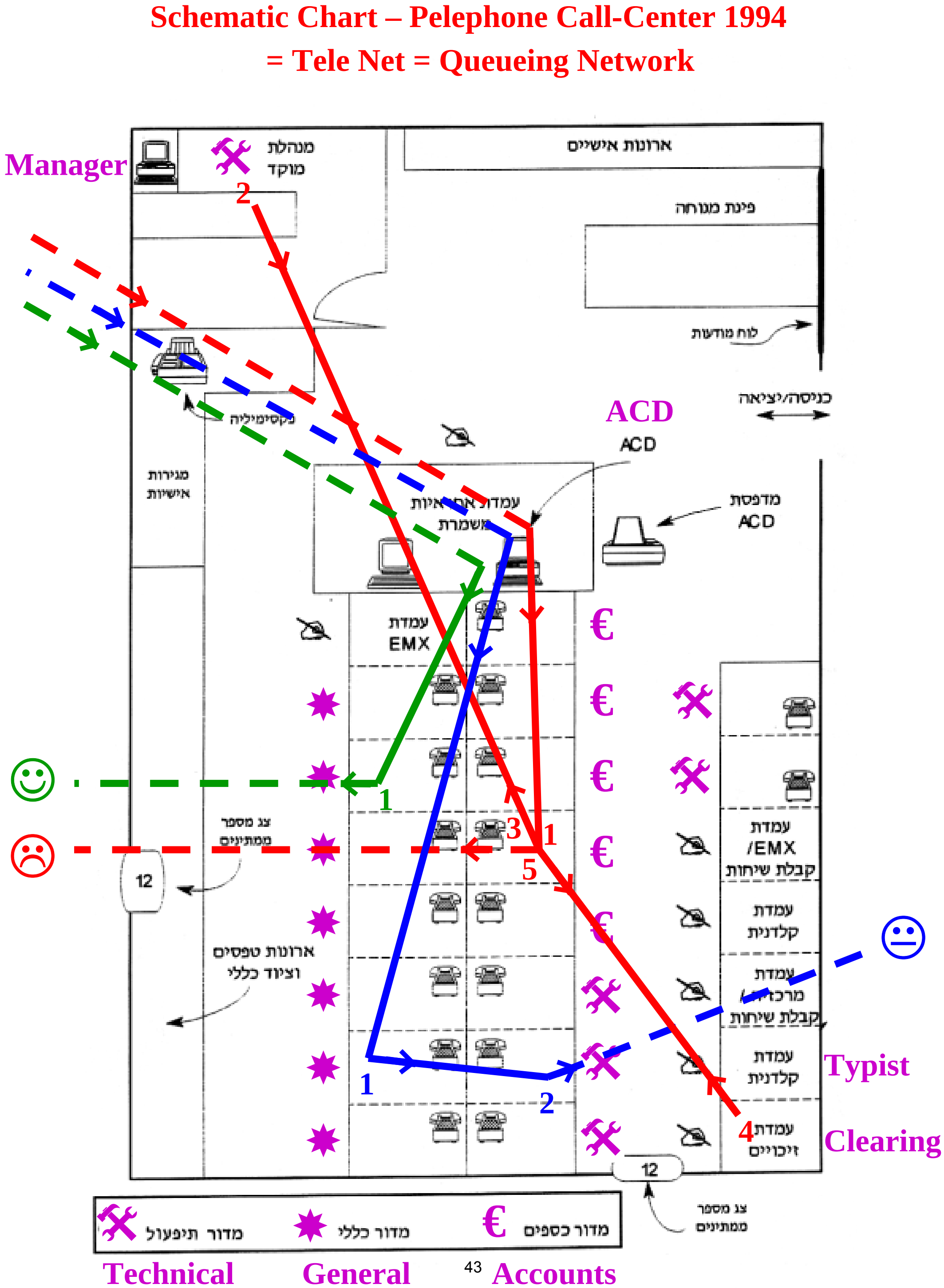
Mapping the Offered Load (Bank Branch)



Note: What can / should be done at 11:00 ?

Conclusion: Models are not always necessary but measurements are !

Conceptual Model: Call-Center Network



43

Conceptual Model: Call-Center Network

Current Status - Analysis

	Accounts Center	General Center	Technical Center
Peak days in a week	Sun, Fri	Sun	Sun
Peak days in a month	12	8-14, 2-3	10-20
Avg. applications no. in a day	4136	2476	1762
Avg. applications no. in an hour - λ_{avg}	**253.6**	**193**	**167**
Peak hours in a day	**11:00-12:00**	**10:00-11:00**	**9:00-10:00**
Avg. applications no. in peak hours - λ_{max}	**422**	**313**	**230**
Avg. waiting time (secs.)	10.9	20.0	55.9
Avg. service time (secs.)	83.5	131.3	143.2
Service index	0.88	0.87	0.72
Abandonment percentage	2.7	5.6	11.2
Avg. waiting time before abandonment (secs.)	9.7	16.8	43.2
Avg. staffing level	9.7	10.3	5.2
Target waiting time	12	25	-

JOINT COMMISSION ON ACCREDITATION OF HEALTHCARE ORGANIZATIONS

2006 HOSPITAL ACCREDITATION STANDARDS FOR
Emergency Management Planning
Emergency Management Drills
Infection Control
Disaster Privileges

(Please note that standards addressing emergency management drills and disaster privileges are undergoing additional research; revised standards for these areas are forthcoming)

Standard EC.4.10
The hospital addresses emergency management.

Rationale for EC.4.10
An emergency[1] in the hospital or its community could suddenly and significantly affect the need for the hospital's services or its ability to provide those services. Therefore, a hospital needs to have an emergency management plan that comprehensively describes its approach to emergencies in the hospital or in its community.

Elements of Performance for EC.4.10
1. The hospital conducts a hazard vulnerability analysis[2] to identify potential emergencies that could affect the need for its services or its ability to provide those services.

2. The hospital establishes the following with the community:
- Priorities among the potential emergencies identified in the hazard vulnerability analysis
- The hospital's role in relation to a communitywide emergency management program
- An "all-hazards" command structure within the hospital that links with the community's command structure

3. The hospital develops and maintains a written emergency management plan describing the process for disaster readiness and emergency management, and implements it when

[1]**Emergency** A natural or manmade event that significantly disrupts the environment of care (for example, damage to the hospital's building(s) and grounds due to severe winds, storms, or earthquakes) that significantly disrupts care, treatment and services (for example, loss of utilities such as power, water, or telephones due to floods, civil disturbances, accidents, or emergencies within the hospital or in its community); or that results in sudden, significantly changed, or increased demands for the hospital's services (for example, bioterrorist attack, building collapse, plane crash in the organization's community). Some emergencies are called "disasters" or "potential injury creating events" (PICEs).

[2] **Hazard vulnerability analysis**: The identification of potential emergencies and the direct and indirect effects these emergencies may have on the hospital's operations and the demand for its services.

4. The business continuity/disaster recovery plan is implemented when information systems are interrupted.

Standard LD.3.15
The leaders develop and implement plans to identify and mitigate impediments to efficient patient flow throughout the hospital.

Rationale for LD.3.15
Managing the flow of patients through the organization is essential to the prevention and mitigation of patient crowding, a problem that can lead to lapses in patient safety and quality of care. The Emergency Department is particularly vulnerable to experiencing negative effects of inefficiency in the management of this process. While Emergency Departments have little control over the volume and type of patient arrivals and most hospitals have lost the "surge capacity" that existed at one time to manage the elastic nature of emergency admissions, other opportunities for improvement do exist. Overcrowding has been shown to be primarily an organization-wide "system problem" and not just a problem for which a solution resides within the emergency department. Opportunities for improvement often exist outside the emergency department.

This standard emphasizes the role of assessment and planning for effective and efficient patient flow throughout the organization. To understand the system implications of the issues, leadership should identify all of the processes critical to patient flow through the hospital system from the time the patient arrives, through admitting, patient assessment and treatment, and discharge. Supporting processes such as diagnostic, communication, and patient transportation are included if identified by leadership as impacting patient flow. Relevant indicators are selected and data is collected and analyzed to enable monitoring and improvement of processes.

A key component of the standard addresses the needs of admitted patients who are in temporary bed locations awaiting an inpatient bed. Twelve key elements of care have been identified to ensure adequate and appropriate care for admitted patients in temporary locations. These elements have implications across the organization and should be considered when planning care and services for these patients. Additional standard chapters relevant to these key elements are shown in parenthesis.

- Life Safety Code issues (for example, patients in open areas) (EC)
- Patient privacy and confidentiality (RI)
- Cross training and coordination among programs and services to ensure adequate staffing, particularly nursing staff (HR)
- Designation of a physician to manage the care of the admitted patient in a temporary location, without compromising the quality of care given to other ED patients (HR)
- Proper technology and equipment to meet patient needs (PC, LD)
- Appropriately privileged practitioners to provide patient care beyond immediate emergency services (HR)

Stochastic Systems

PATIENT FLOW IN HOSPITALS: A DATA-BASED QUEUEING-SCIENCE PERSPECTIVE

By Mor Armony[*], Shlomo Israelit[†], Avishai Mandelbaum[‡], Yariv N. Marmor[§], Yulia Tseytlin[¶], and Galit B. Yom-Tov[‖]

NYU[*], *Rambam Hospital*[†], *Technion*[‡], *Mayo Clinic*[§], *IBM Research*[¶], *Columbia University*[‖]

Patient flow in hospitals can be naturally modeled as a queueing network, where patients are the customers, and medical staff, beds and equipment are the servers. But are there special features of such a network that sets it apart from prevalent models of queueing networks? To address this question, we use Exploratory Data Analysis (EDA) to study detailed patient flow data from a large Israeli hospital.

EDA reveals interesting and significant phenomena, which are not readily explained by available queueing models, and which raise questions such as: What queueing model best describes the distribution of the number of patients in the Emergency Department (ED); and how do such models accommodate existing throughput degradation during peak congestion? What time resolutions and operational regimes are relevant for modeling patient length of stay in the Internal Wards (IWs)? While routing patients from the ED to the IWs, how to control delays in concert with fair workload allocation among the wards? Which leads one to ask how to measure this workload: Is it proportional to bed occupancy levels? How is it related to patient turnover rates?

Our research addresses such questions and explores their operational and scientific significance. Moreover, the above questions mostly address medical units unilaterally, but EDA underscores the need for and benefit from a comparative-integrative view: for example, comparing IWs to the Maternity and Oncology wards, or relating ED bottlenecks to IW physician protocols. All this gives rise to additional questions that offer opportunities for further research, in Queueing Theory, its applications and beyond.

CONTENTS

Keywords and phrases: Queueing Models, Queueing Networks, Healthcare, Patient flow, EDA

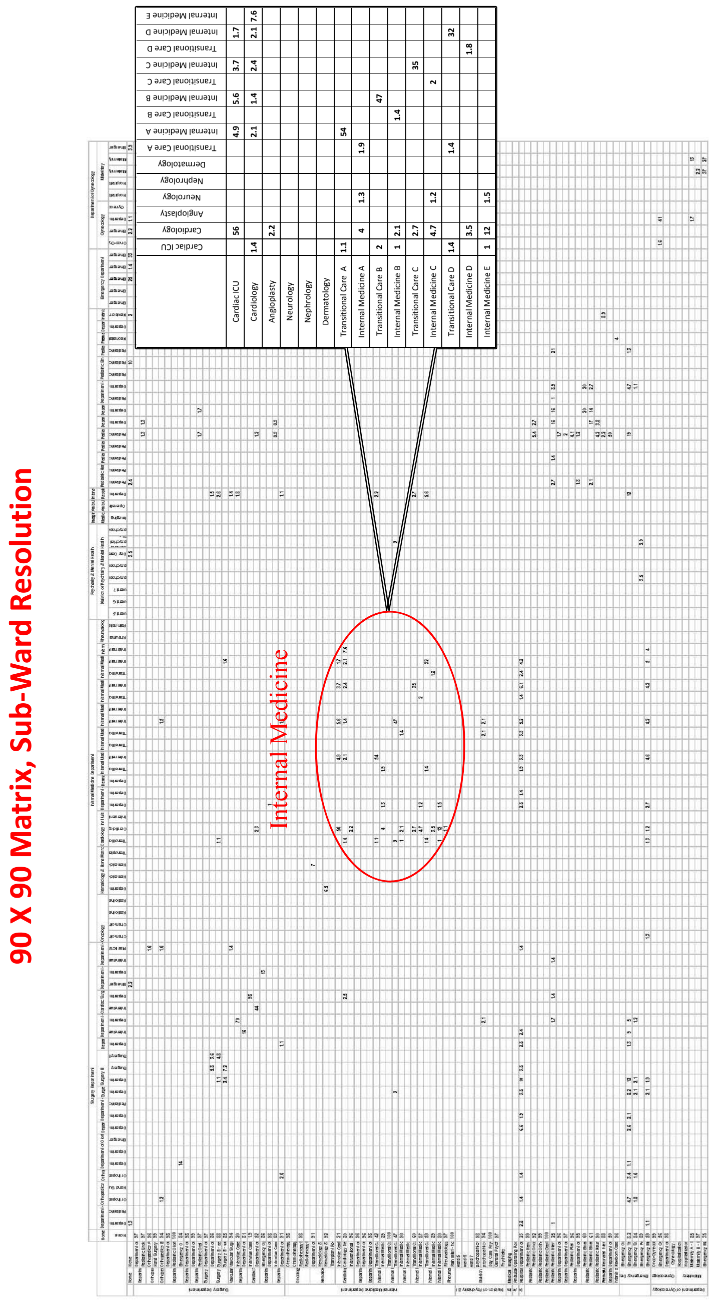


FIG 2. *Transition probabilities between hospital wards, at the resolution of sub-wards. For example, during the period over which the matrix was calculated (January 4th, 2005 to June 31st, 2005), 47% of the patients in the Transitional Care Unit of IW A were transferred to IW A itself. plausibly after their condition improved enough for the transfer.*

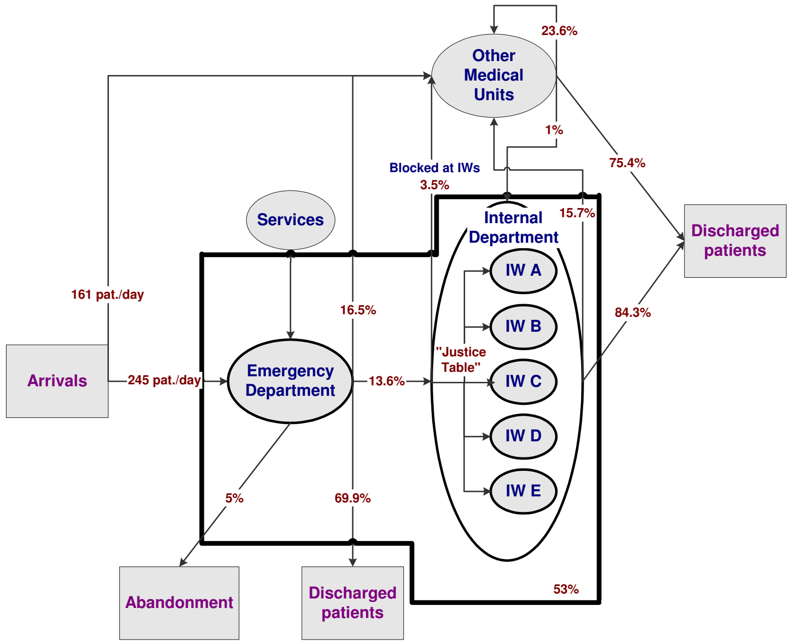


Fig 1. *The ED+IW system as a queueing network*

overall hospital data. One such benefit is the use of other hospital units (e.g. Oncology, Maternity) as reference points. This improves one's understanding of specific phenomena that arise from the ED+IW data.

The ED+IW network. The ED has 40 beds and it treats on average 245 patients daily. An internal patient, whom an ED physician decides to hospitalize, is directed to one of the five Internal wards. The IWs have about 170 beds that accommodate around 1000 patients per month. Internal Wards are responsible for the treatment of a wide range of internal conditions, thus providing inpatient medical care to thousands of patients each year. Wards A-D share more or less the same medical capabilities - each can treat similar (multiple) types of patients. Ward E, on the other hand, attends to only the less severe cases; in particular, this ward cannot admit ventilated patients.

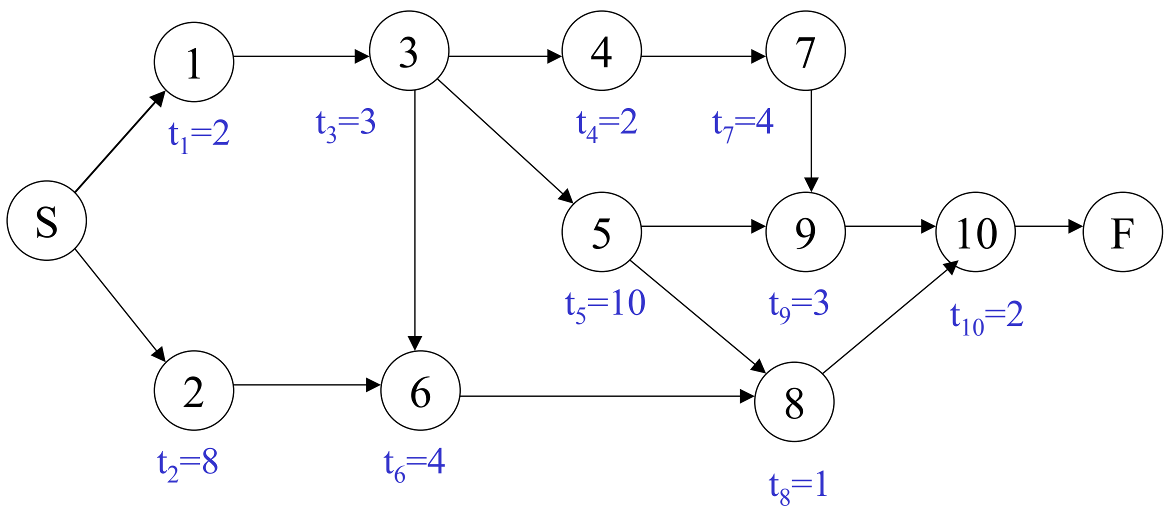
1.3. *Some hints to the literature.* Patient flow in hospitals has been studied extensively. Readers are referred to the many papers in Hall (2006), which are also sources for further references. In the present section, we merely touch on three dimensions, which are the most relevant for our study: a network

Project Management:
Example of Classical Approach

Tennis Tournament Activities
(Fitzsimmons, pp 391–392)

Task Description	Code	Immediate Predecessors
Negotiate for location	1	—
Contact seeded players	2	—
Plan promotion	3	1
Locate officials	4	3
Send invitations	5	3
Sign player contracts	6	2,3
Purchase balls and trophies	7	4
Negotiate catering	8	5,6
Prepare location	9	5,7
Tournament	10	8,9

PERT Chart

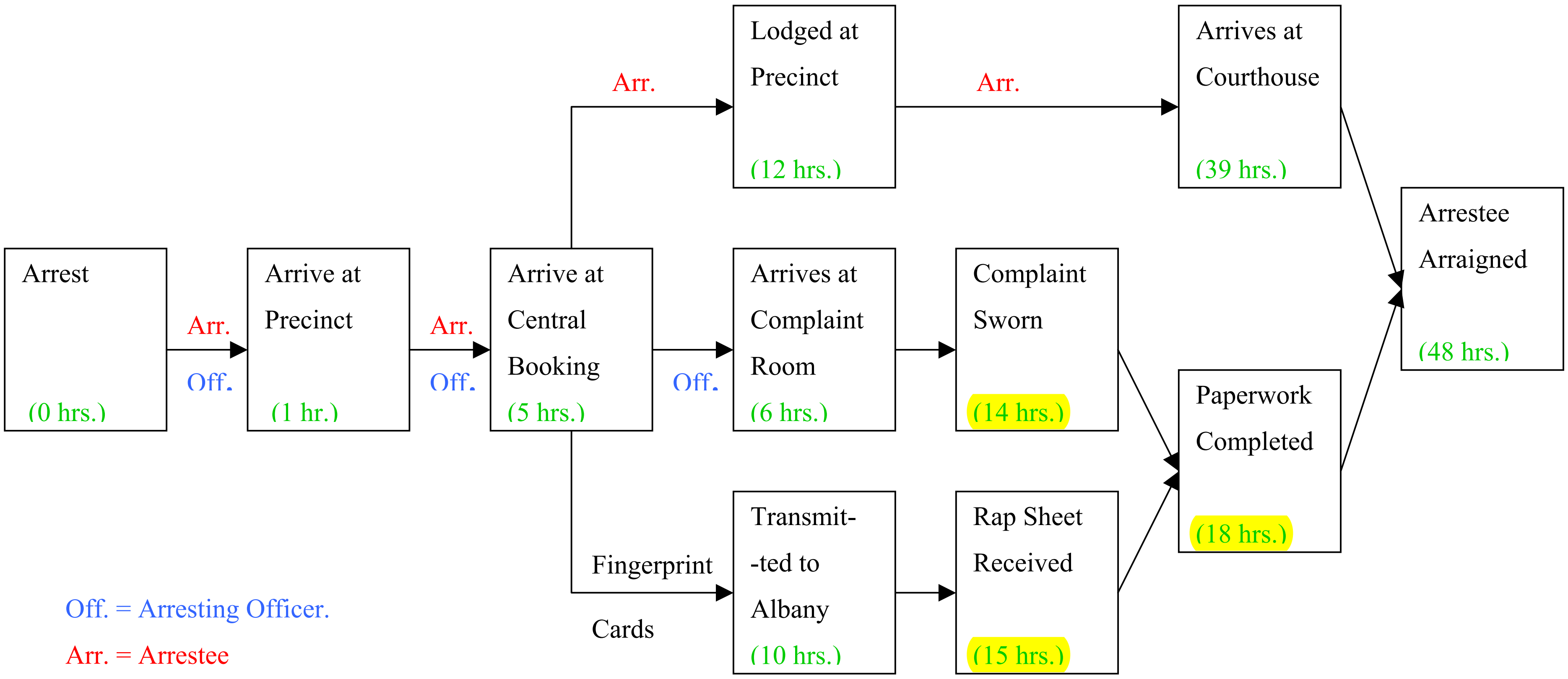


PERT = **P**rogram **E**valuation and **R**eview **T**echnique.

t_i – **completion times** of tasks.
Assume that t_i are **deterministic**.

How to calculate project completion time?

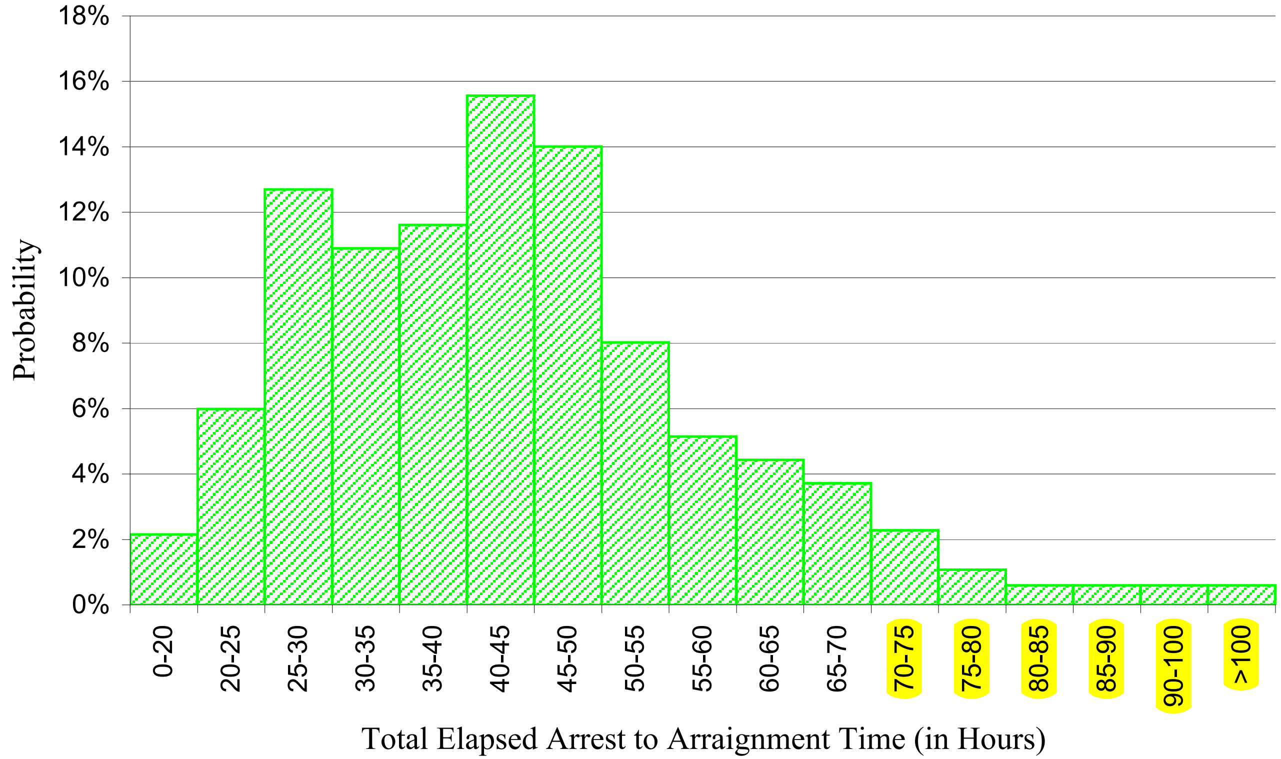
Arrest - to – Arraignment (Larson, ...)



Source:
Improving the N.Y.C A-to-A system
Larson, R. Cahn, M. Shell, M.
Interfaces 23, 1993

Arrest to Arraignment Time

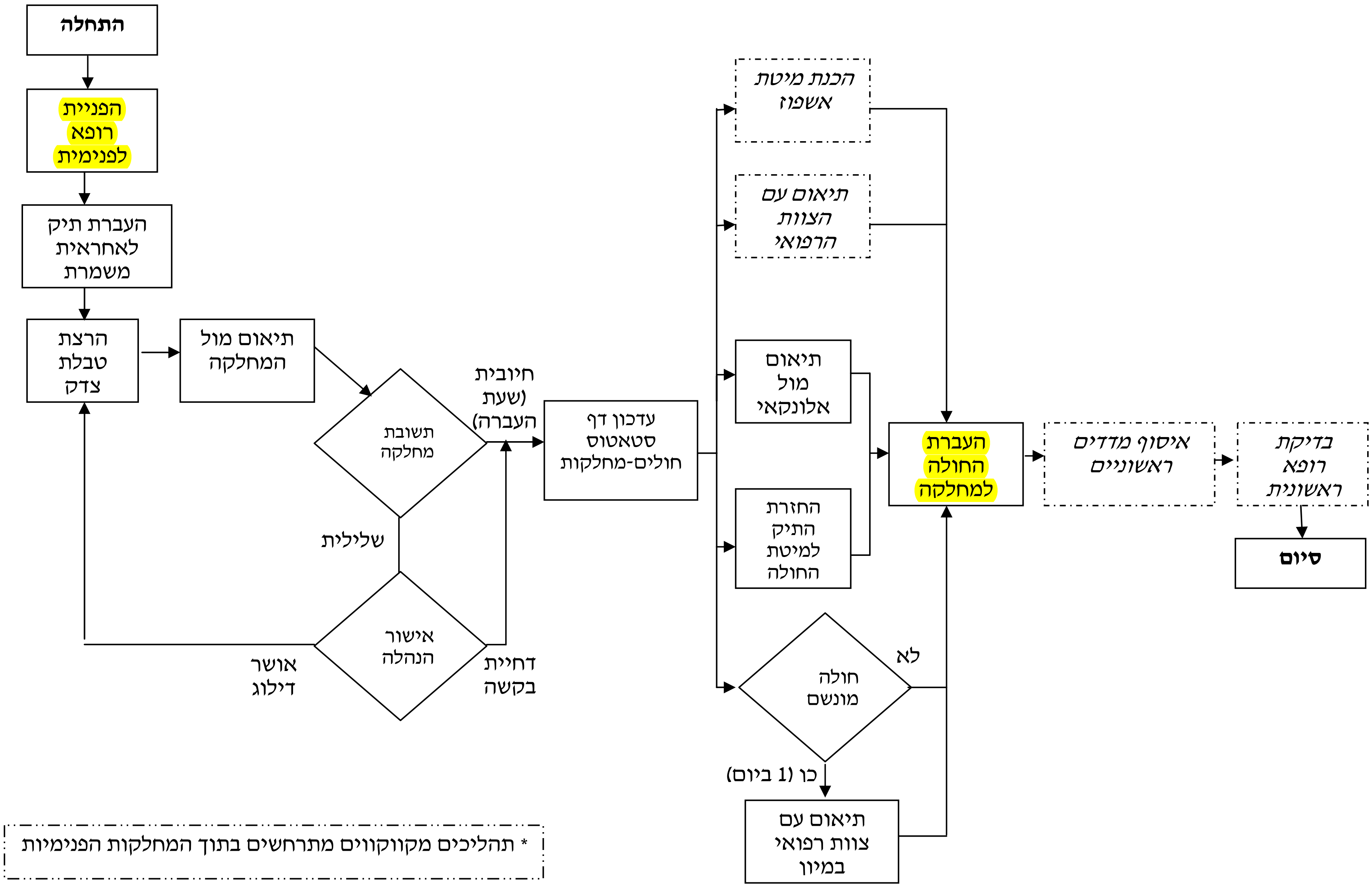
Stochastic dynamic model:



Avg. 44.0 hours
Std. 16.2

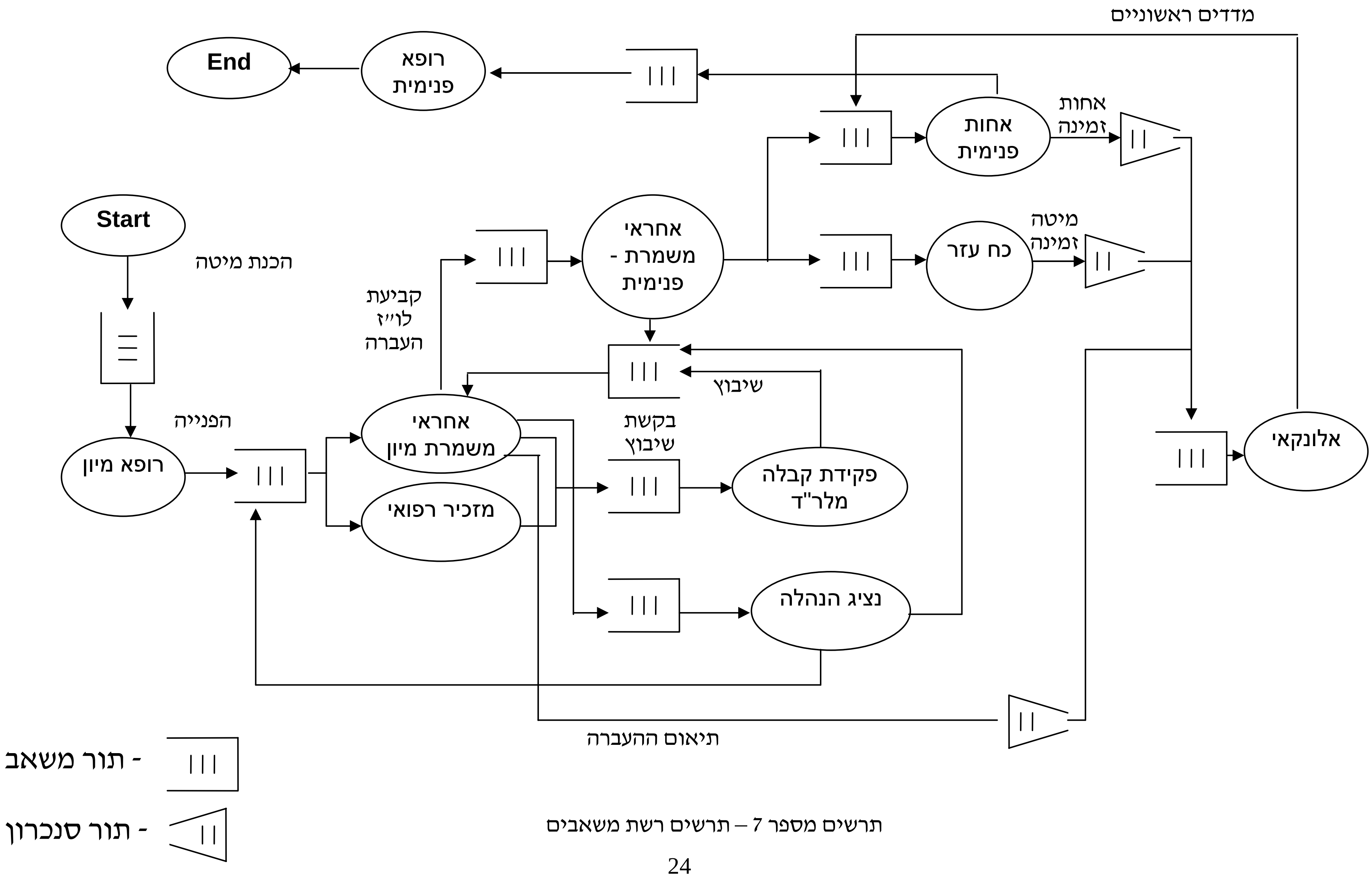
Should be less than 24 hours.

נספח ח' – תרשים פעילויות וקדימויות



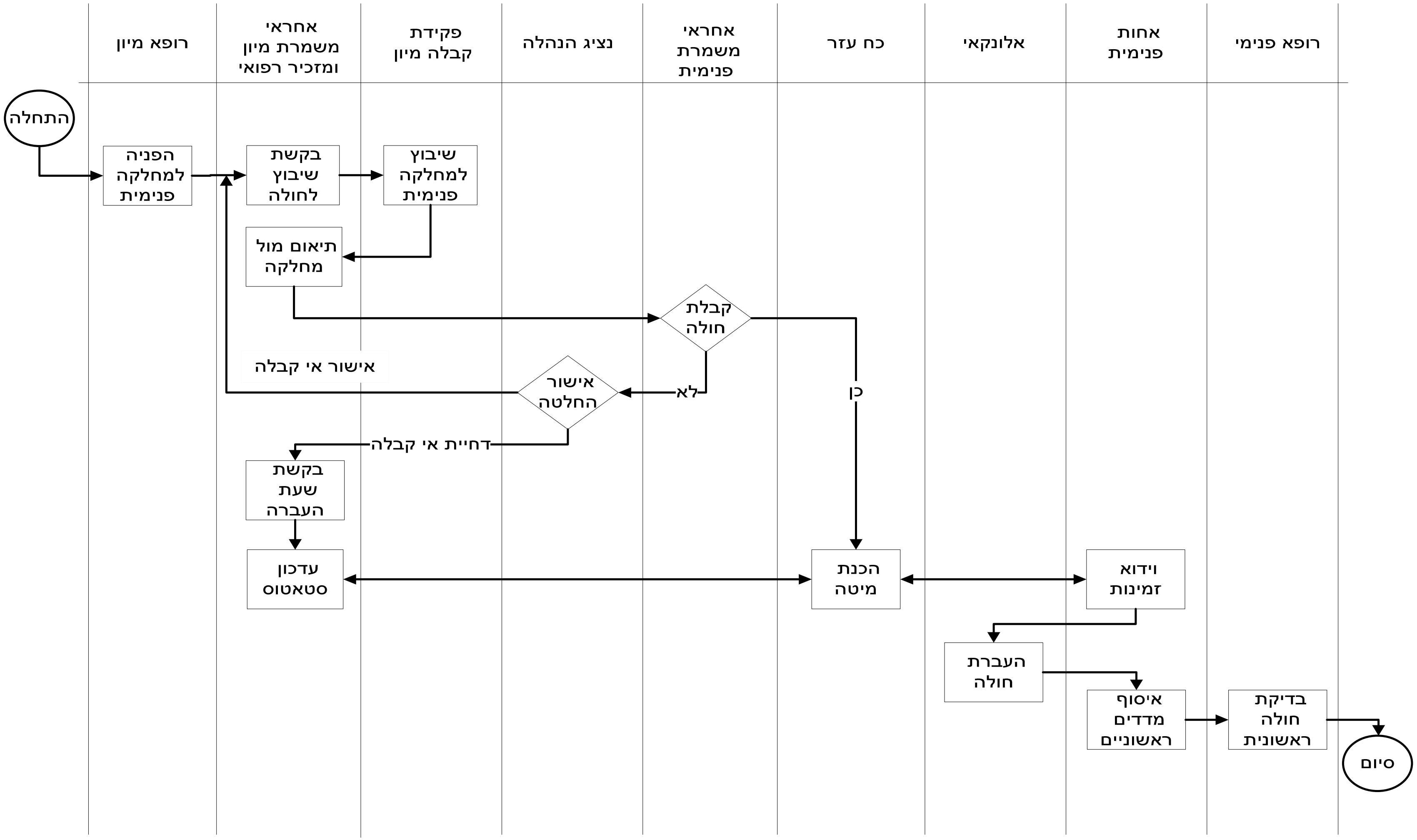
תרשים מספר 6 – תרשים פעילויות וקדימויות

נספח ט' – תרשים רשת משאבים



תרשים מספר 7 – תרשים רשת משאבים

נספח י׳ – תרשים תהליך משולב



תרשים מספר 8 – תרשים תהליך משולב

Why Queues ?
via Dynamic Stochastic PERT/CPM Networks

- Product/Service development
- Project management

Both "enjoy":

  - Stochastic environment
  - Multi-projects
  - Scarce resources

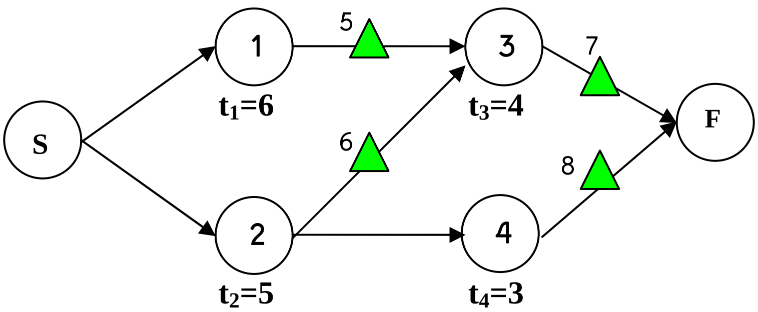
Service Engineering
Created: July 1996
Last revision: November 2003

Dynamic Stochastic PERT/CPM Networks

PERT = **P**rogram **E**valuation and **R**eview **T**echnique (**R**esearch **T**ask);
CPM = **C**ritical **P**ath **M**ethod.

Consider a four-task project, whose precedence constraints are expressed by the network diagram below.
The time required for task i is t_i days on average. There are n_i identical "servers" dedicated to task i, and there are many statistically independent replicas of the project to be completed over time.

Model 1: Deterministic PERT/CPM



Critical path is S-1-3-F.
Project Completion Time is 10 days.

Model 2: Stochastic PERT/CPM

Warmup model: t_1 = 1 or 11, equally likely, which does not alter given averages.
What is then project duration? How about a 13-days deadline? Critical path?

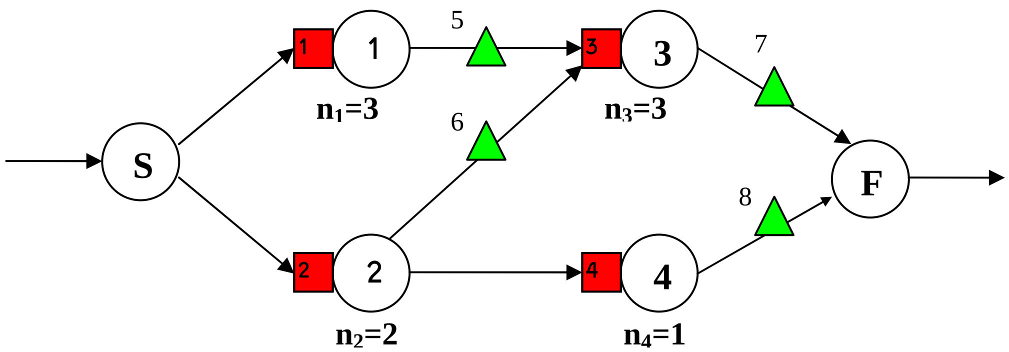
More realistically: Time required for task i is exponentially distributed with mean t_i days and the various task times are independent (random variables). Simulation (spreadsheet) then shows that: Mean completion time = 13.13 days; Standard deviation = 7.36.

Model 3: Dynamic Stochastic Project Networks (PERT/CPM)

New projects are generated according to a Poisson process, the interarrival time being exponential with mean 3.5 days. Each task is processed at a dedicated service station. Tasks associated with successive projects contend for resources on a FIFO basis.
There are two types of queues:
resource-queue: where tasks wait for the resource.
synchronization-queue: where tasks wait until their precedence constraints are fulfilled.

Resource queue (1-4).

Synchronization queue (5-8).



<u>Remark</u>
In general, there could be alternative reasonable definitions of synchronization queues and synchronization times (waiting times in synchronization queues). Such definitions and their interpretations should depend on the particular application in question.
For example, an "Israeli" protocol would specify that if activity 1 is completed before its matching activity 2 then, rather then wait in synchronization queue 5, it immediately joins resource queue 3, waiting there for activity 2 with the hope that it arrives before 1 is admitted to service.

Simulation Description for the Stochastic Models

The behavior of the **static** system was simulated for 50 replications, each with 20,000 projects.
For the static model, project completion time and the distribution of critical path were compiled for each project. (We used 50 replications, each with 20,000 projects, instead of 50*20,000=1,000,000 replications of individual projects, in order to get an approximately normal sample out of the 50 replication means. This is needed to generate confidence intervals, as described further below.)

For the **dynamic** model, the behavior of the system was simulated for 50 replications, 20,000 days each replication.
Data from the first 10,000 days of the simulated operation was discarded, and then summary statistics were compiled for the remaining days of operation.

In both the static and dynamic models, for each project there are three possible critical paths:
s-1-3-f
s-2-3-f
s-2-4-f .
Throughput time, time in queues and critical path were compiled for each project.
Then, for each replication, time statistics were calculated, as well as the distribution of critical paths. These are summarized in subsequent figures, yielding in particular, the mean and std of the throughput time.
At the end of the simulation, standard deviation and confidence intervals were derived, according to the following.

Formulas: ; i- replication index, n= 50.

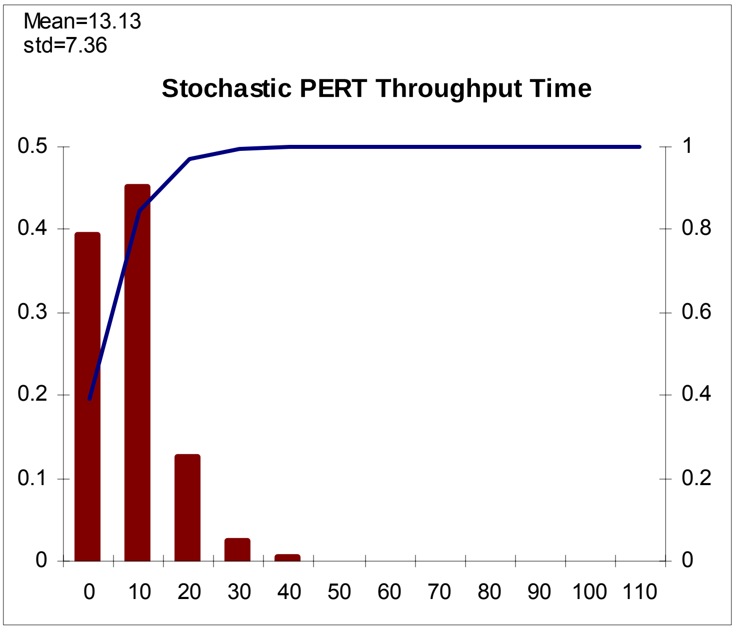
$m = \sum_i x_i / n$; overall mean (x_i - mean of replication i).
$\sigma^2 = \sum_i (x_i - m)^2 / (n-1)$; estimate of variance.
$1-\alpha = 0.95$; confidence level.
$h = t_{n-1,1-\alpha/2} * \sigma / \sqrt{n}$; half-width $1-\alpha$ confidence interval for the mean (based on the normal approximation).

Remark.
The probability that the mean project completion time lies within the interval [m-h, m+h] is equal to $1-\alpha$.

Simulation Results for the Static Stochastic Model

1. Throughput Time

Mean: 13.13 days.
Std: 7.36.   Half C.I: 0.095 (0.46% of the mean).



2. Critical Paths

Path	Frequency	half C.I.
s-1-3-f	0.47	0.0074
s-2-3-f	0.26	0.006
s-2-4-f	0.27	0.0058
	1.00	

3. Critical Activities

Criticality index = the probability that a task is on a critical path.

Task	Criticality index
1	0.47
2	0.53
3	0.73
4	0.27

Results for the Dynamic Stochastic Model

1. Capacity Analysis

Question: Can we do it (in steady state) ?
Answer: Calculate servers' utilization ρ, where ρ = λ*E(S) / n.
The answer is NO – We Can't, if some ρ > 1.

λ - rate of new projects. (And also the processing rate at <u>each</u> of the activity nodes!)
E(S) - mean service time of the station.
n - number of (statistically identical) servers at the station.

Servers' utilization (**%**) = **57,71,38,86**.

2. Response-time Analysis

Question: How long will it take?
Answer: Calculate response/throughput/cycle time.
Present via histograms and Gantt charts.

3. What-if Analysis

Question: Can we do better?
Answer: Sensitivity and parametric analysis.

4. Optimality Analysis

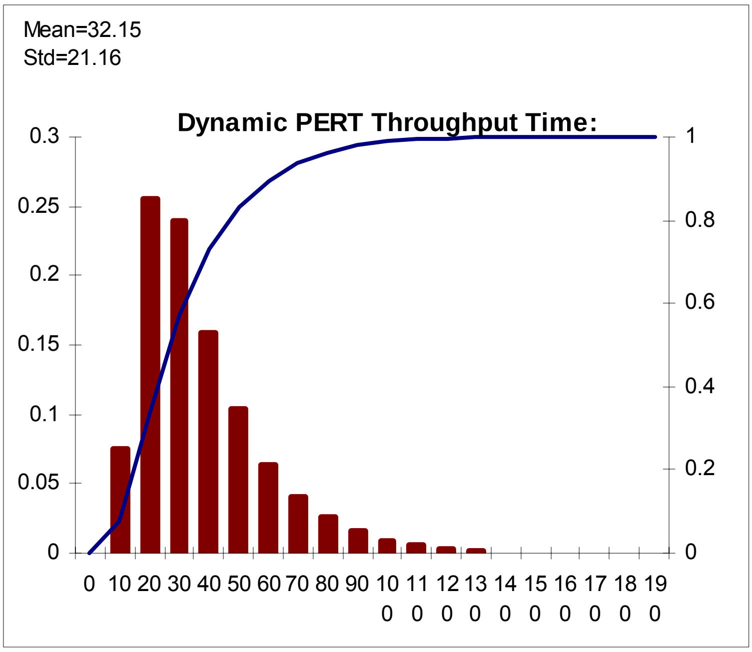
Question: How much better? or: What is the best one could do?
Answer: typically impossible but increasingly possible,
especially in special cases or circumstances.

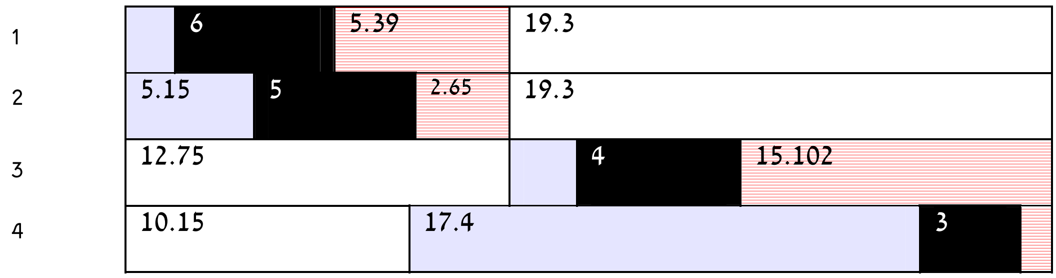
2. Response Time Analysis

How long will it take ? Calculate response/throughput/cycle time.

2.1 Throughput time

Mean=32.15 days.
Std=21.16. Half C.I.=1.5 (4.6% of the mean).
Time Profile: Processing time: 18 days (26.89%).
Waiting time: 24.204 days (36.16%).
Synchronization time: 24.731 days (36.95%).





- waiting time.
- processing.
- synchronization.
- Internal

2.2. Waiting time in queues

Queue	mean	half C.I.	% from mean
1	1.42	0.063	4.43
2	5.15	0.318	6.17
3	0.234	0.009	3.84
4	17.4	1.49	8.5
5	5.39	0.298	5.5
6	2.65	0.076	2.86
7	15.102	1.42	9.5
8	1.589	0.071	4.46

2.3 Critical paths

Path	Frequency	half C.I.
s-1-3-f	0.146	0.0067
s-2-3-f	0.104	0.0052
s-2-4-f	0.750	0.0110
	1.000	

2.4. Critical tasks

Task	Criticality index
1	0.146
2	0.854
3	0.250
4	0.750

Note that task 2 has, by far, the highest criticality index. Yet, task 4 is the clear bottleneck, as far as waiting time is concerned.
The reason for the former is that task 2 participates in "most" paths of the network (2 out of 3).
A reasonable procedure to identify a "critical task" seems to be as follows:
a. Identify the critical path of maximum likelihood (based on 2.3).
b. Identify the task of maximum waiting time (based on 2.2).

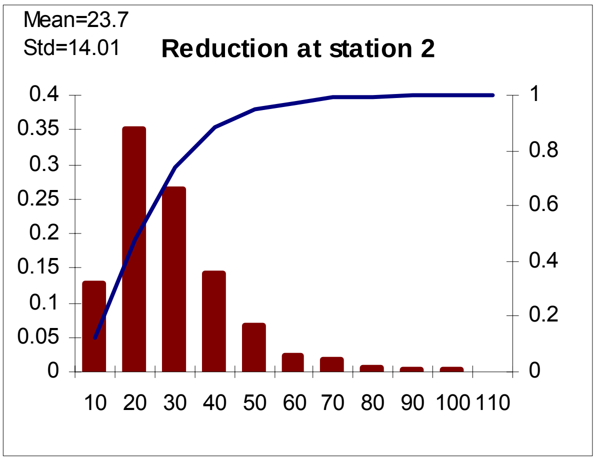
3. What-if Analysis

Question: Can we do better?
Answer: Sensitivity and parametric analysis.

3.1 Reduction at Station 2

Change the mean service time at station 2 to 4 days (instead of 5).
New Mean=23.7 days (improvement of 26.2%).
Std=14.01. Half C.I.=0.368 (1.5% of the mean).
Servers' utilization (%)= 57,57,38,86.

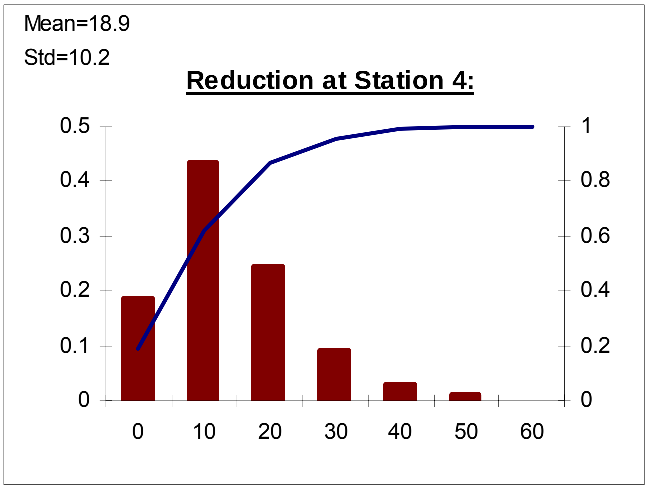
Time Profile: Processing time: 17 days (28.19%).
Waiting time: 21 days (34.82%).
Synchronization time: 22.3 days (36.98%).



3.2 Reduction at Station 4

Change the mean service time at station 4 to 2 days (instead of 3).
New Mean=18.9 days (improvement of 41.2%).
Std=10.2. Half C.I.=0.205 (2% of the mean).
Servers' utilization (%)= 57,71,38,57.

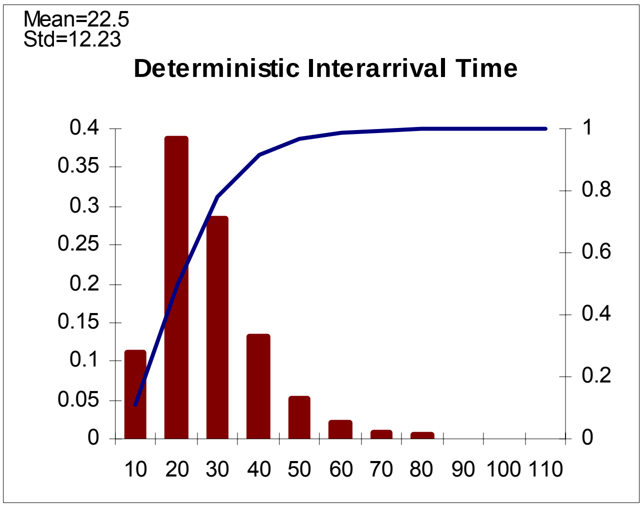
Time Profile: Processing time: 17 days (46%).
Waiting time: 10.6 days (22%).
Synchronization time: 14.5 days (32%).



3.3 Deterministic arrival of projects

Change interarrival time of new projects to exactly 3.5 days (from exponential).
New Mean=22.5 days (improvement of 32.2%).
Std=12.23. Half C.I.=0.63 (2.8% of the mean).
Servers' utilization (%)=57,71,38,86.

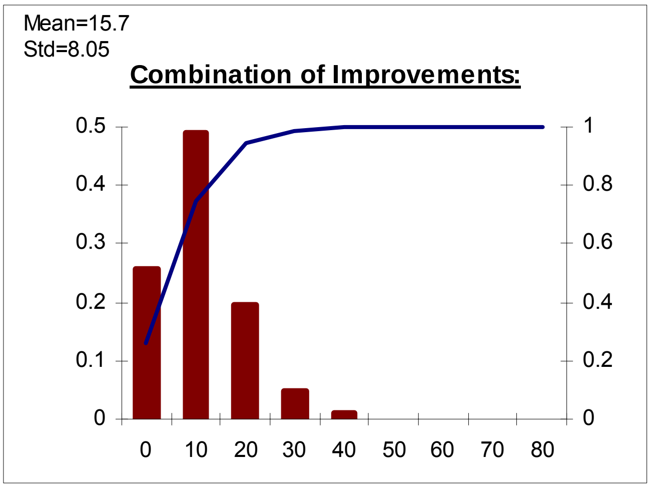
Time Profile: Processing time: 18 days (37.5%).
Waiting time: 12.9 days (26.87%).
Synchronization time: 17.1 days (35.63%).

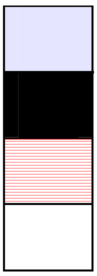


3.4 Combination

Note that a large amount of time is spent at resource queue 4.
Comparing the utilization of station 3 and 4, this suggests a potential process improvement: shift a server from station 3 to 4.
Therefore, the last scenario combines the two improvements: a deterministic interarrival time and shifting one server from station 3 to 4.

New Mean=15.7 days (improvement of 51.16%).
Std=8.05. Half C.I.=0.198 (2% of the mean).
Servers' utilization (%)= 57,71,57,43.
Time Profile: Processing time: 18 days (52.38%).
Waiting time: 3.66 days (10.6%).
Synchronization time: 12.7 days (36.96%).





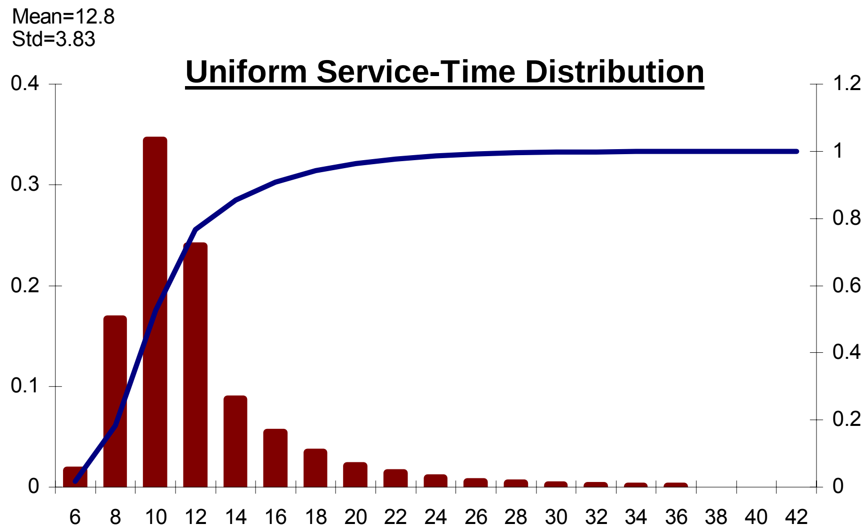
- waiting time.
- processing.
- synchronization.
- internal.

4. Dependence On Distribution

Change, for instance, the distribution of service times from exponential to uniform, but maintain the same mean values as before. Specifically, the time required for task i is uniformly distributed between limits a_i and b_i days, and the interarrival times being uniformly distributed between zero and seven days here.

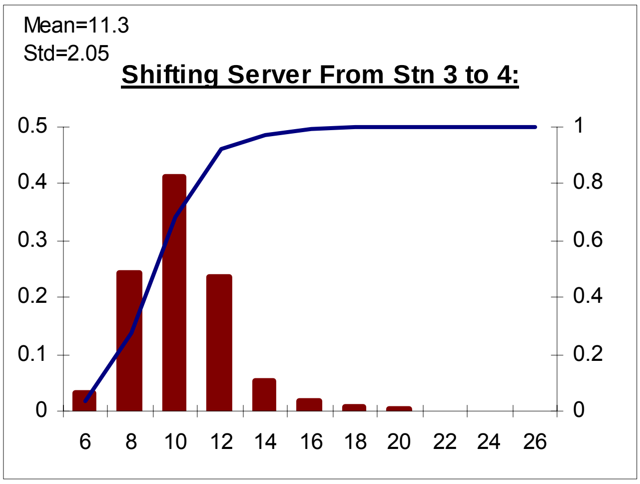
Task	a_i	b_i
1	3	9
2	3	7
3	3	5
4	2	4

New Mean=12.8 days (Compare with 32.15 days in exponential times).
Std=3.83.   Half C.I.=0.034 (0.26% of the mean).
Servers' utilization (%)= 57,71,38,86.



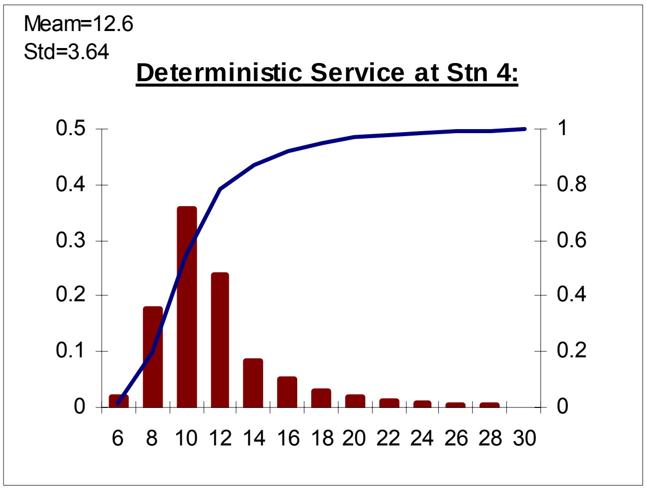
Additional scenarios:

4.1 Allocating Man Power Resources:
Shift a server from station 3 to 4.
New Mean= 11.3 days (improvement of 11.7%).
Std=2.05.   Half C.I.=0.017 (0.15% of the mean).
Servers' utilization (%)=57,71,57,43.



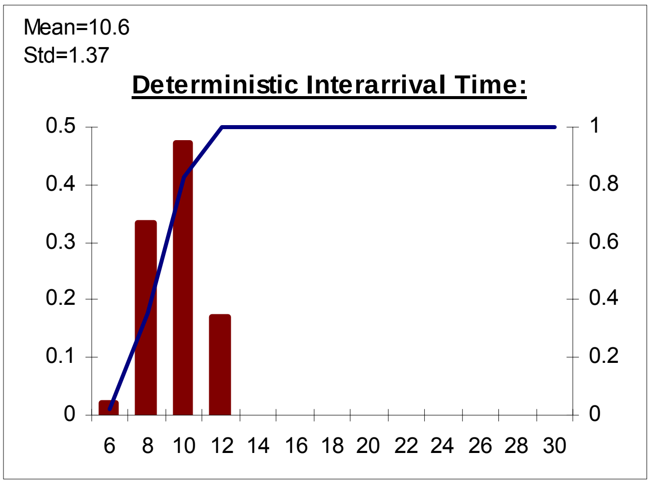
4.2 "TQM" at station 4

Change service at station 4 to deterministic (3).
New Mean=12.6 days (improvement of 1.5%).
Std=3.64. Half C.I.=0.105 (0.83% of the mean).
Servers' utilization (%)=57,71,38,86.



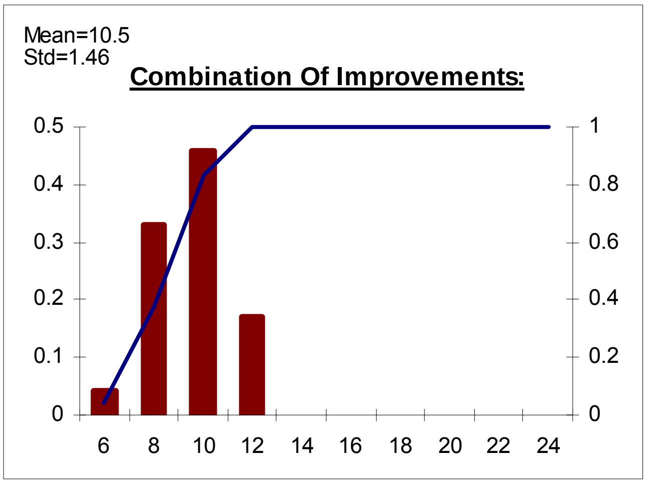
4.3 Deterministic arrival of projects:

Change interarrival time of new projects to exactly 3.5 days.
New Mean=10.6 days (improvment of 17.18%).
Std=1.37. Half C.I.=0.007 (0.066% of the mean).
Servers' utilization (%)=57,71,38,86.



4.4 Combination:

Deterministic interarrival time and shifting one server from station 3 to 4.
New Mean=10.5 days (improvement of 17.96%).
Std=1.46. Half C.I.=0.005 (0.33% of the mean).
Servers' utilization (%)= 57,71,57,43.



5. Dynamic Stochastic Control: Project Management

Consider two types of controls: **open** control, under which all candidate projects are actually initiated, and **closed** where projects must adhere to some predefined criteria in order to be started.
Our open controls are No-Control and MinSLK; the closed control is QSC.
Here are their details:

***Open Controls*:**
1. No Control: A push system with FCFS (First Come First Served) queues. This case was analyzed previously in 2.1.
2. MinSLK: Highest priority in queue to a Minimum Slack activity (MinSLK). Slack-time of an activity is the difference between its Late-Start and Early-Start times. Under MinSLK, as a particular project is delayed then the priorities of its activities increase. Specifically, when an activity of a project is completed, the project's prevalent critical-path is re-evaluated and slack times are updated for the rest of the projects' activities. Then, activities with the least slack time are given the highest priority in resource allocation.

***Closed Control*:**
3. QSC: Queue Size Control (QSC) is based on controlling the resource queue of the bottleneck, the latter being the resource that essentially determines the system's processing capacity. Specifically, one predetermines a maximal number of activities that is allowed, at any given time, within the bottleneck's resource-queue. An arriving project is then allowed into the system to be processed if the length of the bottleneck's resource queue is below this maximal number; otherwise, the arriving project is discarded, never to return (or, alternatively, return late enough so as not to introduce dependencies into the arrival process).

Outline of experiments:

1. **Response time analysis**

2. **The control effect for high throughput rate**

3. **Congestion curves**

5.1. Response time Analysis

How long will it take?
Calculate response/throughput/cycle time.

5.1.1 No Control - see 2.1 on page 6, where we had: Mean = 32.15 days, Std = 21.16.

5.1.2 MinSLK

Mean=21.59 days.
Std=11.57. Half C.I.=0.37 (1.71% of the mean).
Time Profile: Processing time: 18 days (39%).
Waiting time: 12.01 days (26%).
Synchronization time: 16.10 days (35%).

5.1.3 QSC (6)

The maximal number of activities allowed, at any given time, within the resource queue of the bottleneck is 6. (We shall retain this threshold subsequently as well.)

The **bottleneck** resource, namely the resource that determines the system's processing capacity, is taken to be Resource 4. It can be justified by observing that a mere single Resource 4 is dedicated to Task 4; its anticipated utilization level of about 3/3.5 = 86% in steady state, which is by far the highest among all the resources. This choice also finds ample support in our previous analysis (e.g., see 2.2-2.3 on page 7).

λ_{eff} = 0.27 (vs. arrival rate = 0.29 = 1/3.5: 6.9% of the projects are **lost**)

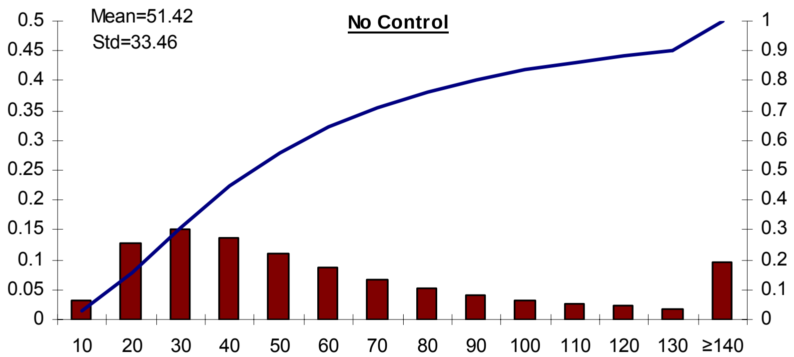
Mean=18.62 days (13.8% lower then MinSLK)
Std=8.80 (23.94% lower then MinSLK). Half C.I.=0.13 (0.70% of the mean).
Time Profile: Processing time: 18 days (54%).
Waiting time: 8.42 days (11%).
Synchronization time: 13.73 days (35%).

5.2. The control effect in heavy traffic

Suppose that the projects arrival rate increases from 1/3.5=0.29 projects/days to an arrival rate of 1/3.25=0.31 projects/days. The load on Resource 4, our bottleneck, increases from 3/3.5=86% to 3/3.25=92%. With this **increased** load, performance is as follows:

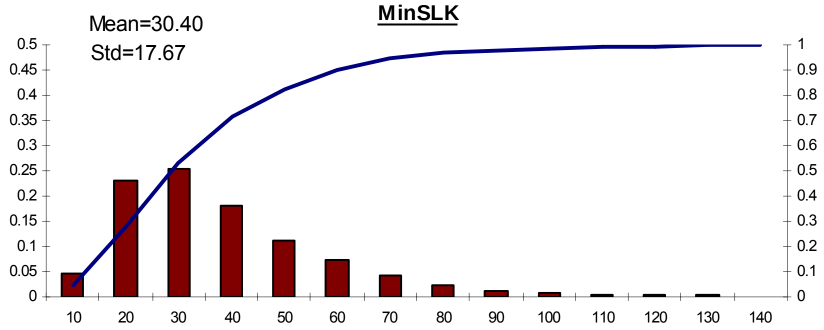
5.2.1 No Control

Mean=51.42 days (vs. 32.15 days in base case).
Std=33.46. Half C.I.=3.86 (7.5% of the mean).
Time Profile: Processing time: 18 days (17%).
Waiting time: 44.83 days (43%).
Synchronization time: 42.42 days (40%).



5.2.2 MinSLK

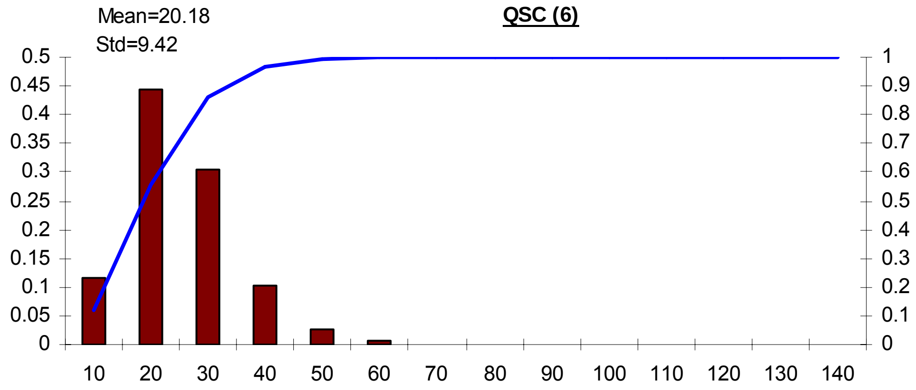
Mean=30.40 days.
Std=17.67. Half C.I.=0.88 (2.9% of the mean).
Time Profile: Processing time: 18 days (27%).
Waiting time: 25.94 days (38%).
Synchronization time: 23.72 days (35%).



5.2.3 QSC (6)

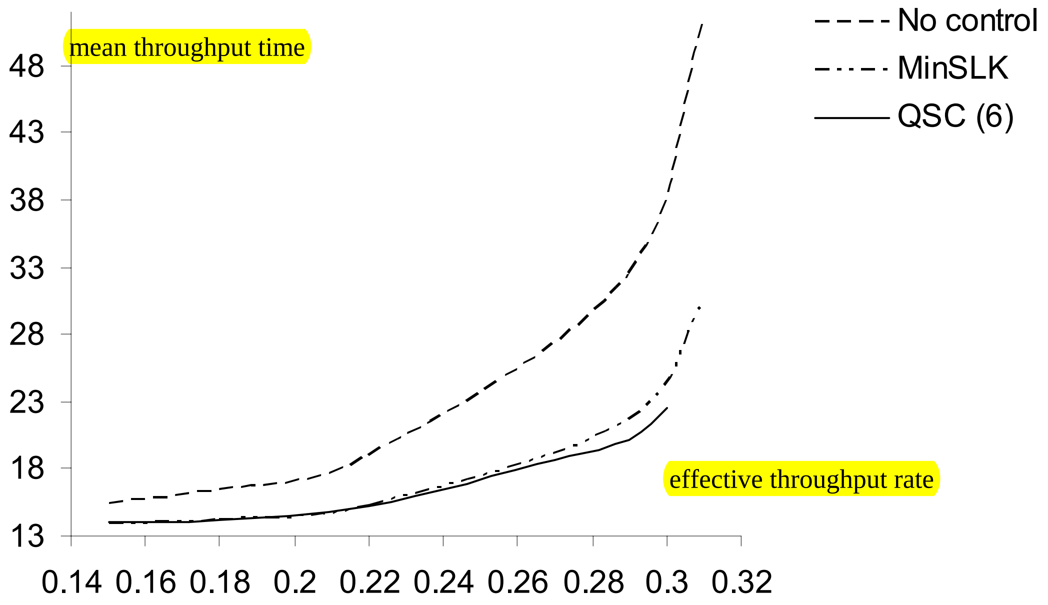
λ_{eff}=0.29 (arrival rate=0.31, 6.4% of the projects are lost)
Mean=20.18 days (33.62% lower then MinSLK; vs. 13.8% under moderate loads.)
Std=9.42 (46.69% lower then MinSLK). Half C.I.=0.19 (0.94% of the mean).

Time Profile: Processing time: 18 days (42%).
Waiting time: 10.47 days (24%).
Synchronization time: 14.67 days (34%).



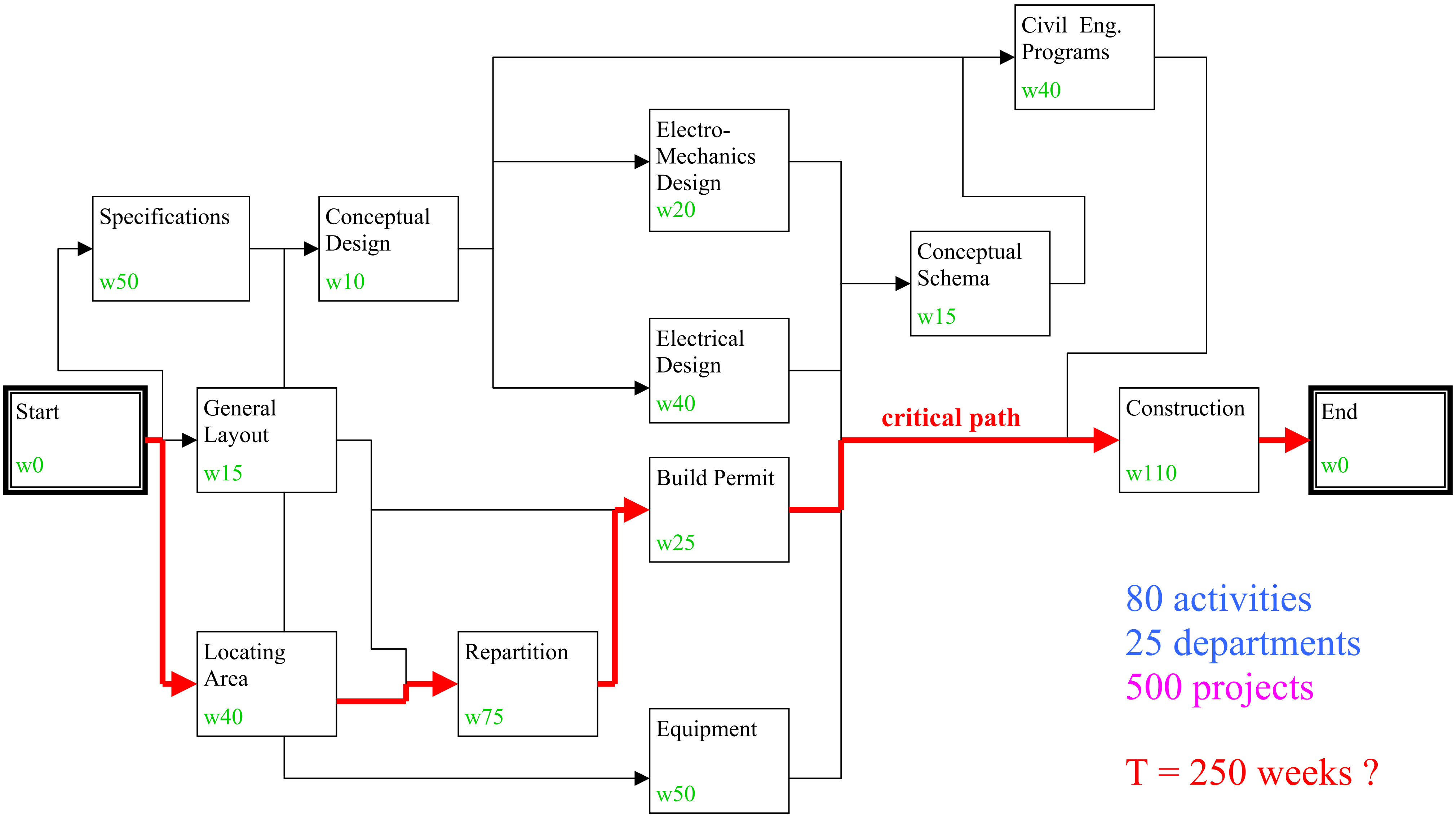
5.3. Congestion Curves

Now we change the effective throughput rate (x-axis) while recording the mean throughput time (y-axis), for throughput rates between 0.14 to 0.32.
The results, for each of our three controls, are as follows:



Traditional PERT/CPM Representation

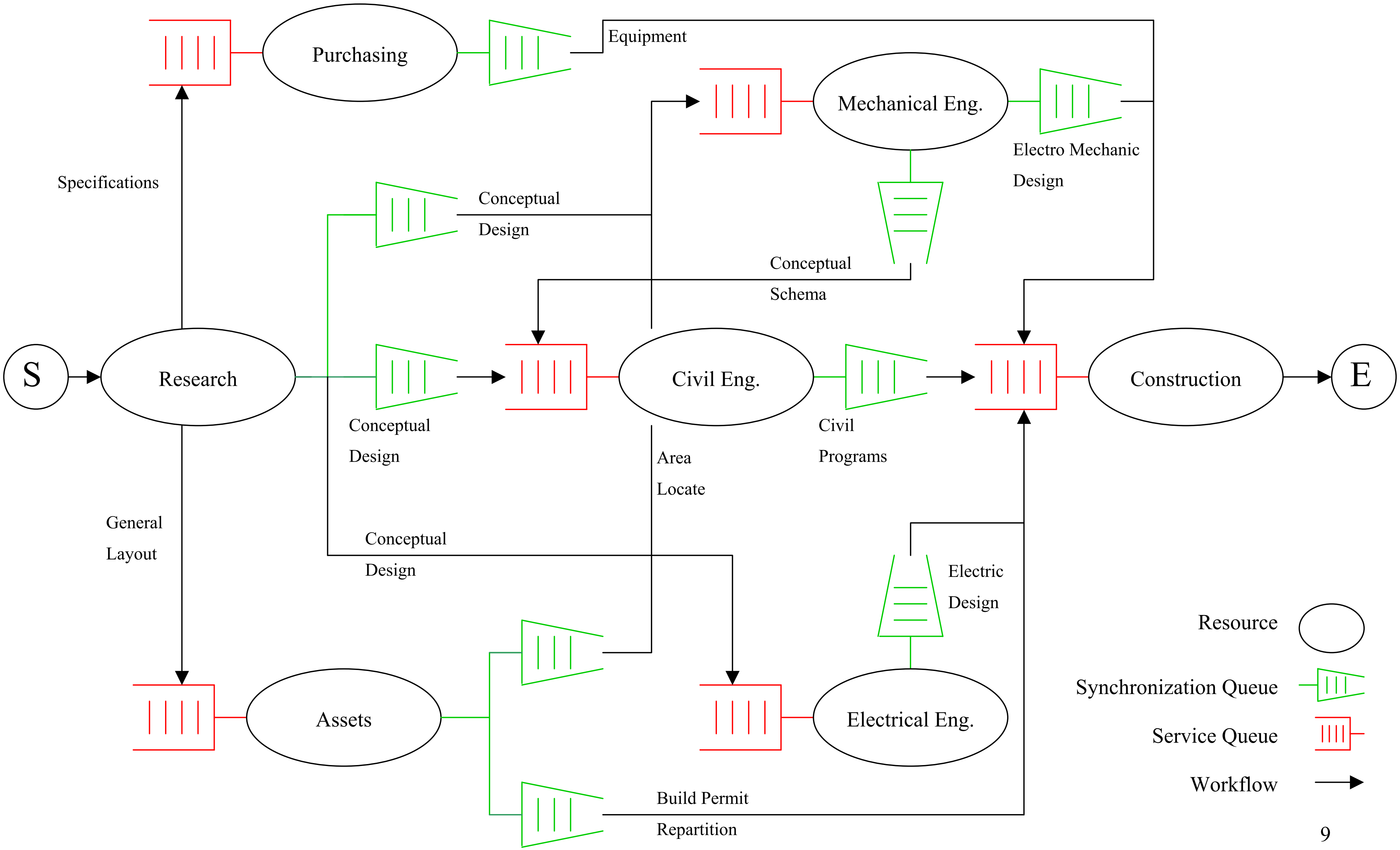
Project View



80 activities
25 departments
500 projects

T = 250 weeks ?

Processing Network Representation

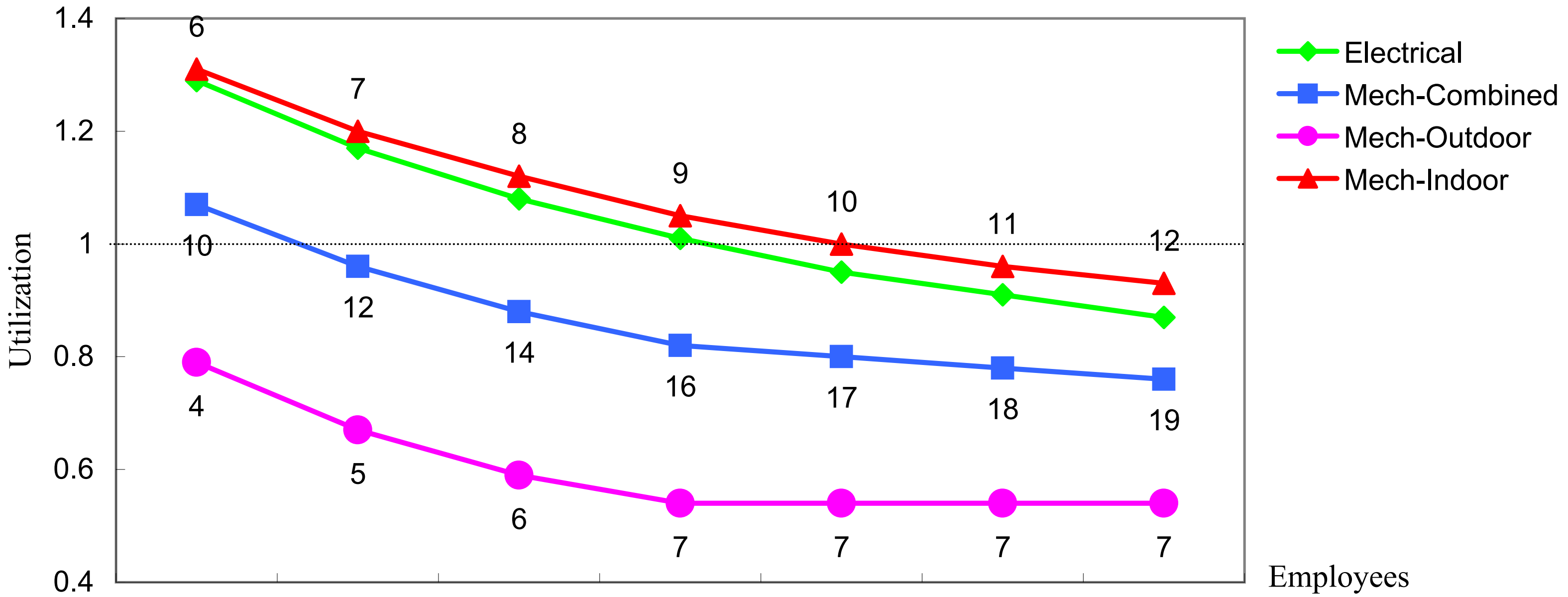


Can We Do It ?

Capacity Analysis -
[= Fluid-view (first moments)]

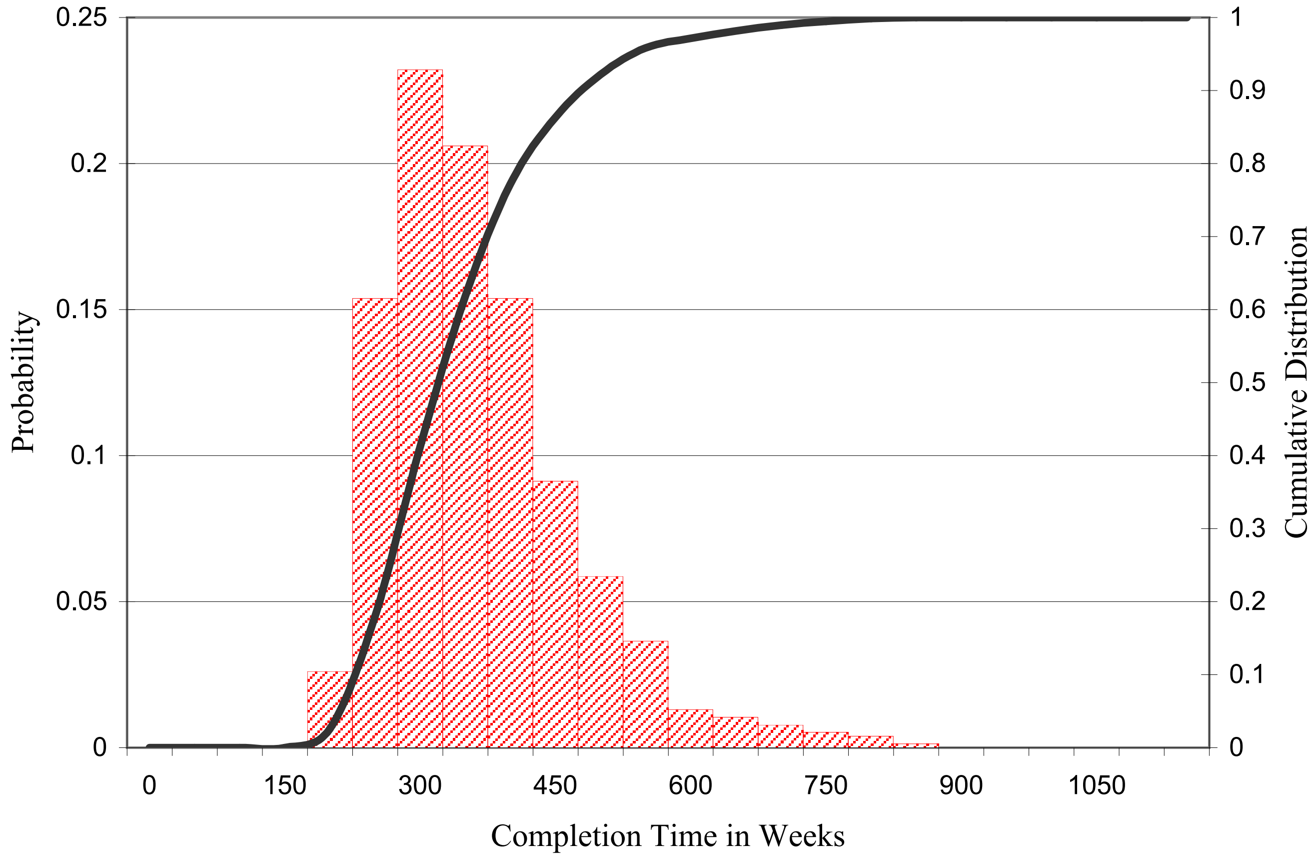
Utilization as a function
of the number of employees:

Electrical		Electro-Mechanic Combined		Electro-Mechanic Outdoor		Electro-Mechanic Indoor	
Employees	Utilization	Employees	Utilization	Employees	Utilization	Employees	Utilization
6	129%	10	107%	4	79%	6	131%
7	117%	12	96%	5	67%	7	120%
8	108%	14	88%	6	59%	8	112%
9	101%	16	82%	7	54%	9	105%
10	95%	17	80%	7	54%	10	100%
11	91%	18	78%	7	54%	11	96%
12	87%	19	76%	7	54%	12	93%



How long Will It Take ?

Stochastic static model (single project):



Avg. 338 weeks

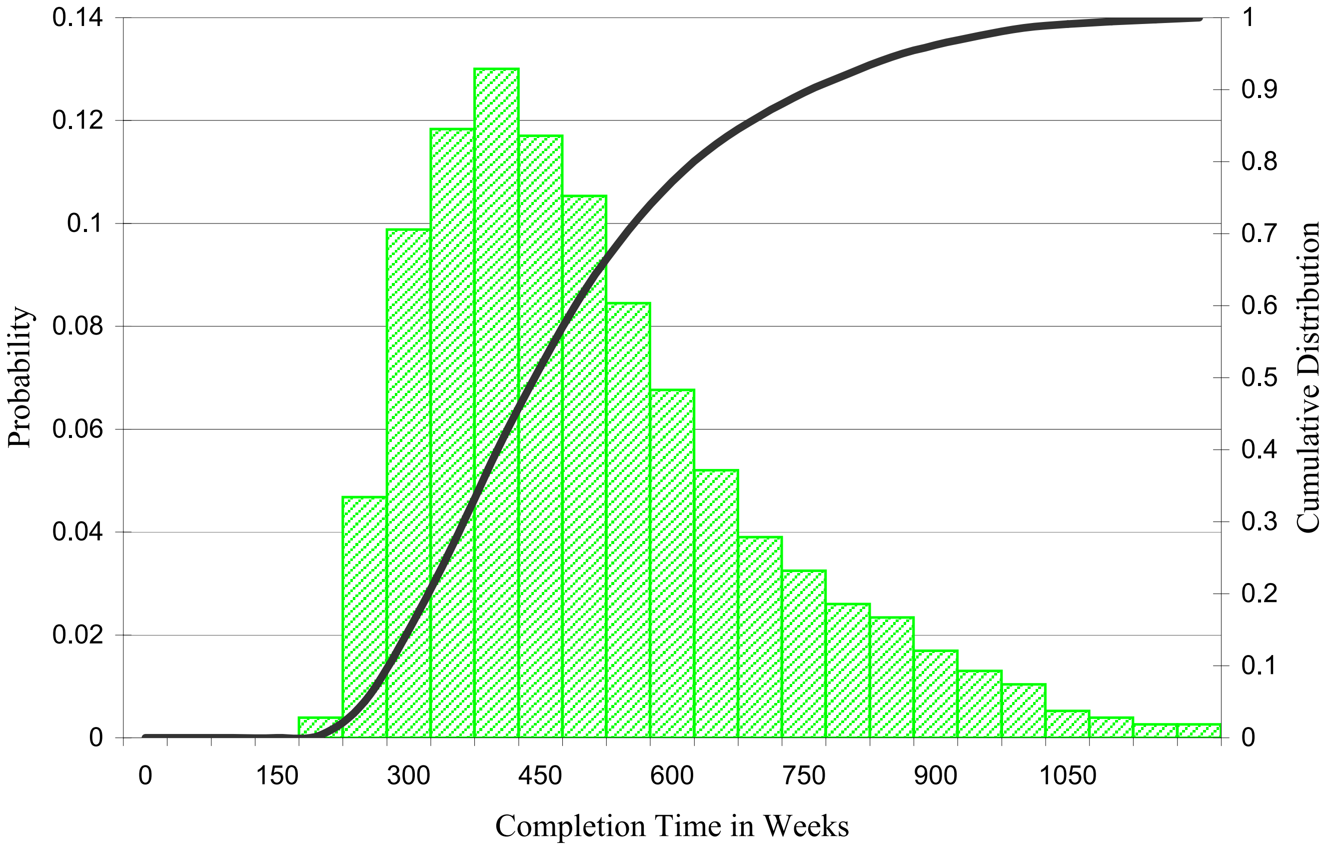
Std. 100

90% 475

Det. 250

How Long Will It Take ?

Stochastic dynamic model:



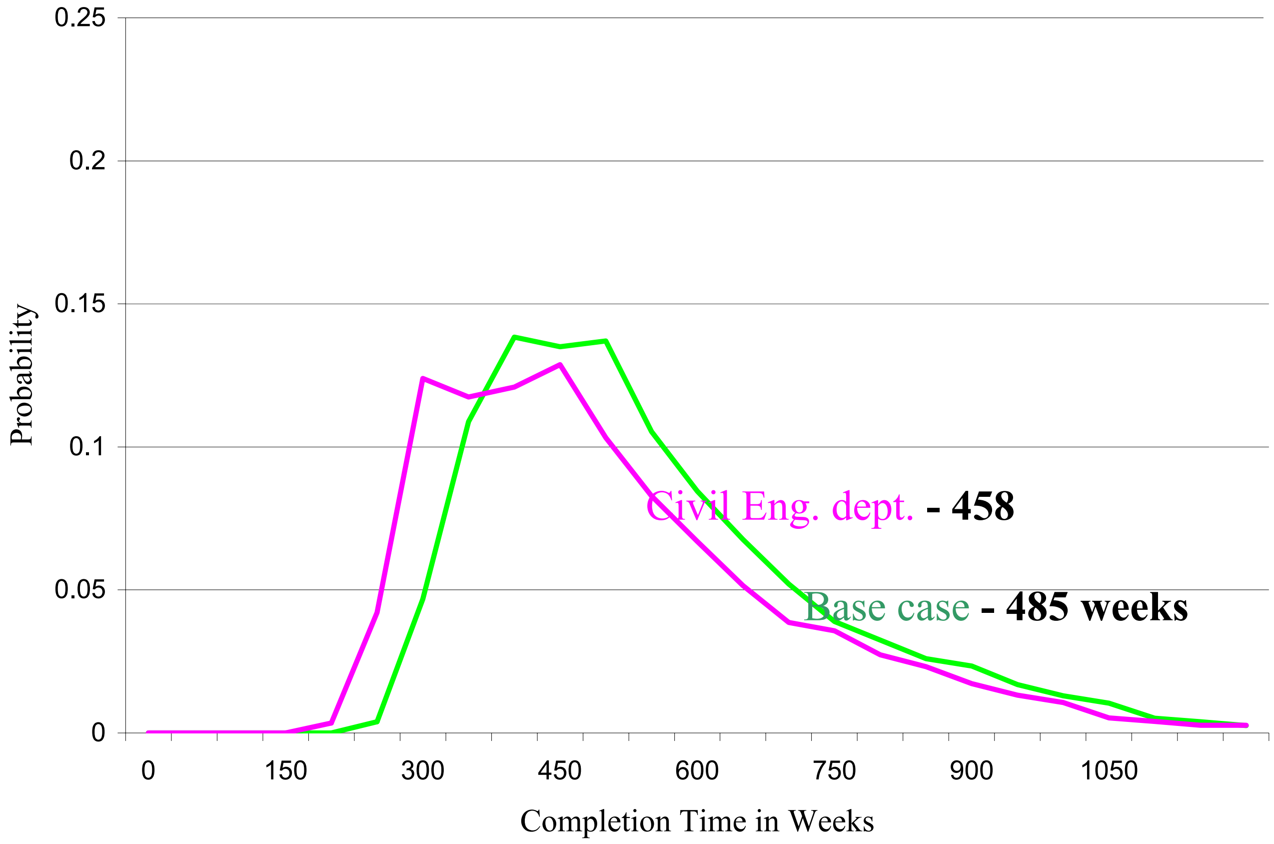
4 Types:

Type	Per year
New sub-station	3.27
New switching stations	0.6
Improvements	3.4
Additional capacity	1.9

Avg.	485 weeks
Std.	199
90%	770
Det.	250

Can We Do Better ?

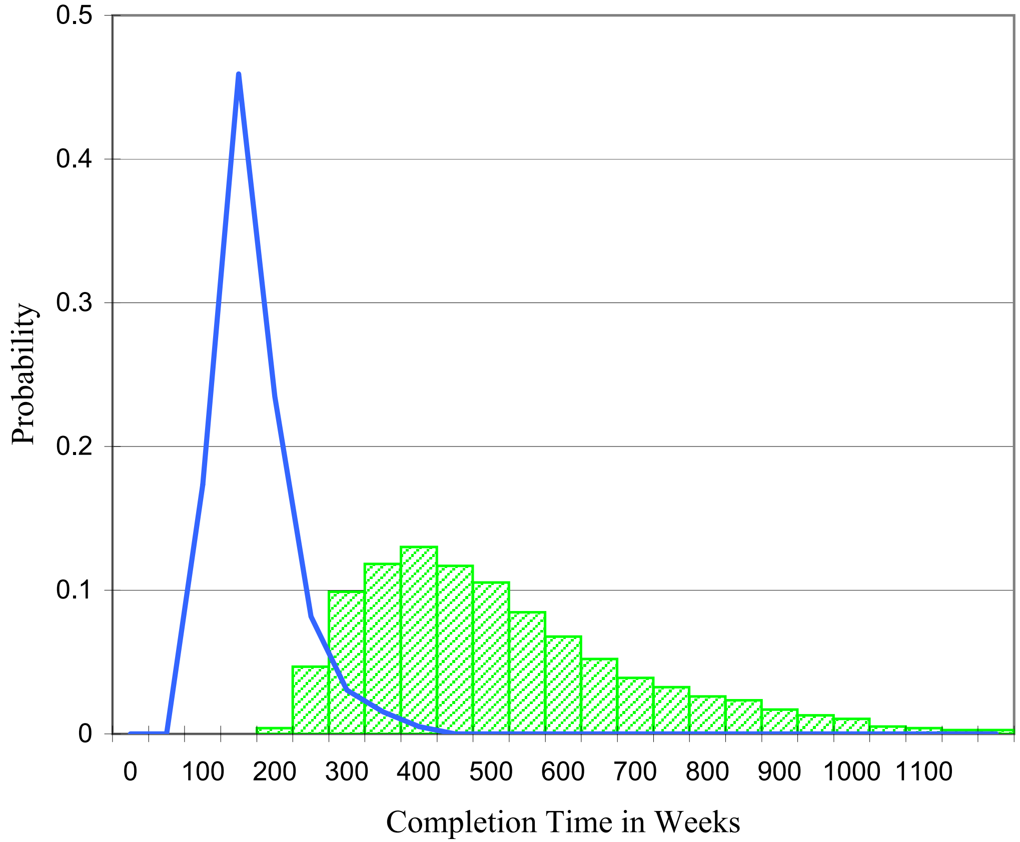
Relieving bottlenecks:



1 2 Reset

Can We Do Better ?

New location management and standardization:



	Base	New
New location mgt:		
	40 weeks,	8 weeks
	0.5 prob. of repeat	0.8 prob.
Standardization:		
	8000 hrs. planning,	2000
	repeats,	none
	long execution times	↓25%
Avg.	485 weeks	189
Std.	199	55
90%	770	294

Summary

	E	σ	90%
Deterministic	251 weeks	0	251
Stochastic Static Single-Project	338	100	475
Stochastic Dynamic Multi-Projects	485	200	770 (14 years)
Infinite Resources	338	...	...
Re-Engineering	189	55	294