Business analytics I
Operations Analytics

Marat Salikhov

March 10th, 2022

About me

- Assistant Professor of Operations
- MAE’14, PhD INSEAD’19, postdoc Yale’21
- Now jointly appointed at MSM Skolkovo and NES
- Questions I’m working on:
  - How to forecast uncertain demand for new products?
  - How to analyze forecasting experts’ performance?
  - How to match workers to customers on platforms?

What is business analytics?

The science of using **data** to build **models** that lead to better **decisions** that add **value** to individuals, to companies, to institutions.

Descriptive

What's going on now?
Examples
Customer segments
Performance evaluations

Predictive

What will happen in the future?
Examples
Demand forecasts
Disruption risk assessment

Prescriptive

What should be done?
Examples
Pricing
Inventory management

Who uses it?

How much of a new collection to order?

How many nurses to staff?

How to price the tickets?

Course logistics

Parts of the course

- Two parts
- My part: Operations Analytics
  - Questions:
    - How much productive capacity do we need to have?
    - How much inventory to order?
    - How to match workers to orders?
  - Focus on optimization
- Daria’s part: Marketing Analytics
  - Questions
    - Which attributes do customers value?
    - How valuable is a customer for a company?
    - How can we cluster customers in groups?
  - Focus on data analysis

How will our course be structured?

- Four mini-modules
  - **Linear and integer optimization**
    - How to make optimal decisions?
  - **Probabilistic modeling and simulation**
    - How to account for uncertainty?
  - **Stochastic optimization**
    - How to make optimal decisions when things are uncertain?
  - **Dynamic optimization**
    - Adding time into the mix

How will each module be structured?

- Mini-module
  - **Motivation:** MBA-style class. How can we apply it
  - **Problem statement:** How to translate into rigorous math?
  - **Implementation:** How to solve these problems with code?
- At the end of the mini-module
  - Solve an assignment
  - Before the assignment: two sessions
    - Session 1: I explain the assignment to you
    - Session 2: You ask me what is still unclear

Assignments

- Several business problems
- You will need to:
  - Write down a model
  - Solve it and report the results
- Grading: are your results correct?
- Working on assignments: **individually**
  - Can share results, can’t share code
- Submit the results to me on my.nes: code + report

What you need to know

- Basics of programming
  - Preferably in Python
- Some linear algebra and calculus
- Basics of probability theory

Grading and deadlines

5%: class attendance and participation

	Home Assignments	Final project
March 28th	HA 1 deadline **(8%)**	
April 11th	HA 2 deadline **(8%)**	Submit a team of 2 or 3 people. If you don’t have a team, I will assign a team to you at random.
April 25th	HA 3 deadline **(8%)**	Submit a three-paragraph project proposal
May 5th	HA 4 deadline **(8%)**	
May 12th (tentative)		Final project presentations **(13%)**

Midterm project structure

- 13% of the final grade (26% if you’re only taking my half)
- Work in teams
  - Submit the list to me by April 11th
  - No less than 2 members, no more than 3
- What will you need to do?
  - **Propose** a business problem
  - **Formalize** its solution as an optimization problem
  - **Implement** a prototype solution given some example data
  - **Explore** numerically how it behaves
- Submit a proposal to me by April 25th

Midterm project: coming up with ideas

- Take a problem in a home assignment and relax assumptions
- Combine the ideas from two different sub-modules
- Come up with a problem that interests you personally
- Ask me for an idea (but then do a great implementation)
- Or just do what you want!

Midterm project grading

- How can you gain points for the midterm project?
  - **Creativity:** does the idea stand out?
  - **Formalization:** is the model either original or technically sophisticated?
  - **Implementation:** are you able to solve the problem to optimality?
  - **Exploration:** did you evaluate your model? On how many dimensions?

Questions?

Class 1
Process Analysis

Motivation: Process Analysis

- How do we come up with an efficient way to do something?
- By considering a process

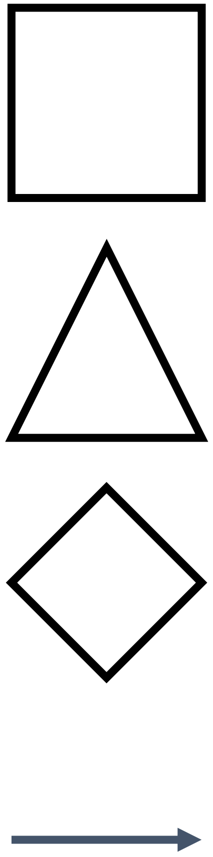
- A process is any transformation from inputs to outputs

- Examples:
  - A minting press. Inputs: metal circles. Outputs: coins.
  - McDonalds. Inputs: buns and patties. Outputs: burgers.

- How do we understand a process?
  - By looking inside it and representing it as a sequence of smaller processes connected to each other
  - Main idea: the slowest link drives the entire sequence

- As the unit of input moves along, it is gradually transformed to a unit of output
  - We call it a flow unit while it is inside the process
  - For now, we treat all flow units as interchangeable

Process flow diagram: elements



Activity/Task

Inventory buffer

Decision point

Material flow

Analyzing a process: a recipe

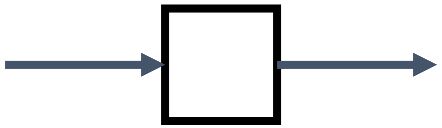
1. Write down the process flow diagram. Determine:
   A. The inputs and the outputs
   B. The tasks and their sequence
   C. Resources used for each task
   D. Where is inventory kept in the process

2. Bottleneck analysis. Determine:
   A. The capacity of each resource and the demand rate
   B. The bottleneck resource
   C. The capacity of the process

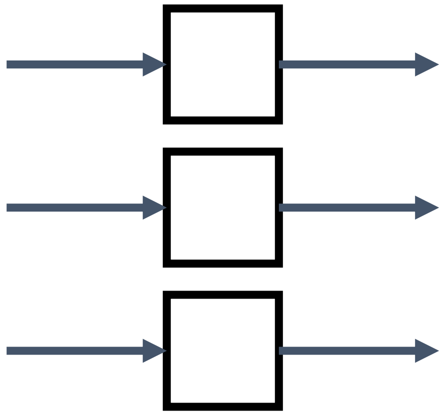
3. Improvement decisions
   A. Recommend on how to improve the system

The process flow diagram: single-stage

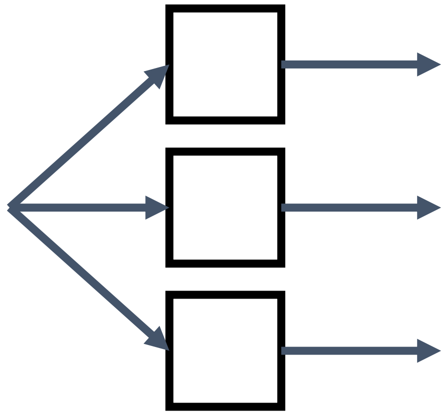
Single station



Parallel stations
Separate queues

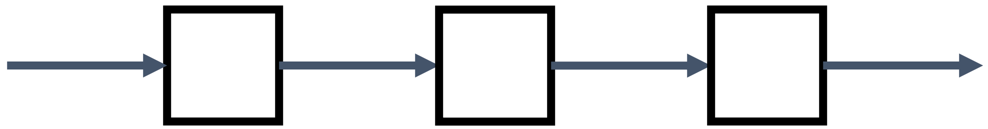


Parallel stations
Single queue

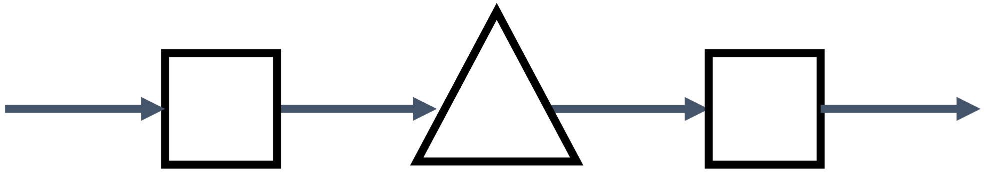


The process flow diagram: multistage

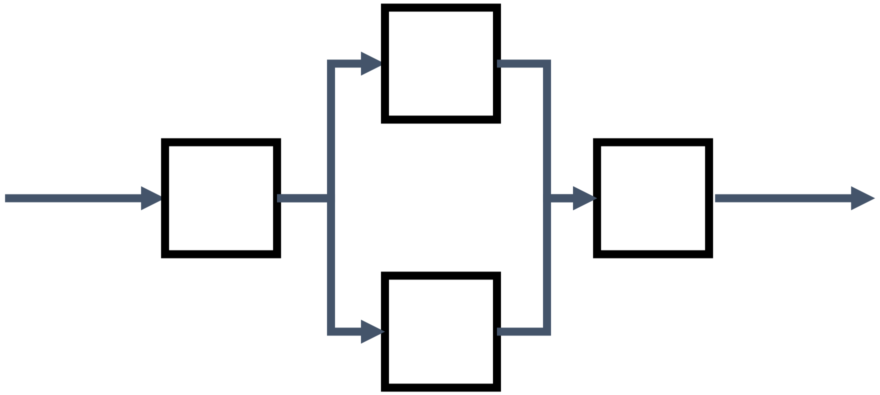
Serial process



Serial process with inventory buffer



Mixed process




ПРИВИВОЧНЫЙ

Example: Form 086/y

- You need this form to be enrolled at an institute of higher learning
- Need to get a referral from your therapist and go through at least
  - Surgeon’s check
  - Ear-nose-throat (ENT) check
  - Eye check
- A mental health evaluation is not always required; but we’ll still consider it
- Assume you already got the referral and managed to schedule all the specialists on one day
- How would the process look like? Let’s find out!

Analyzing a process: a recipe

1. Write down the process flow diagram. Determine:
   A. The inputs and the outputs
   B. The tasks and their sequence
   C. Resources used for each task
   D. Where is inventory kept in the process

2. Bottleneck analysis. Determine:
   A. The capacity of each resource and the demand rate
   B. The bottleneck resource
   C. The capacity of the process

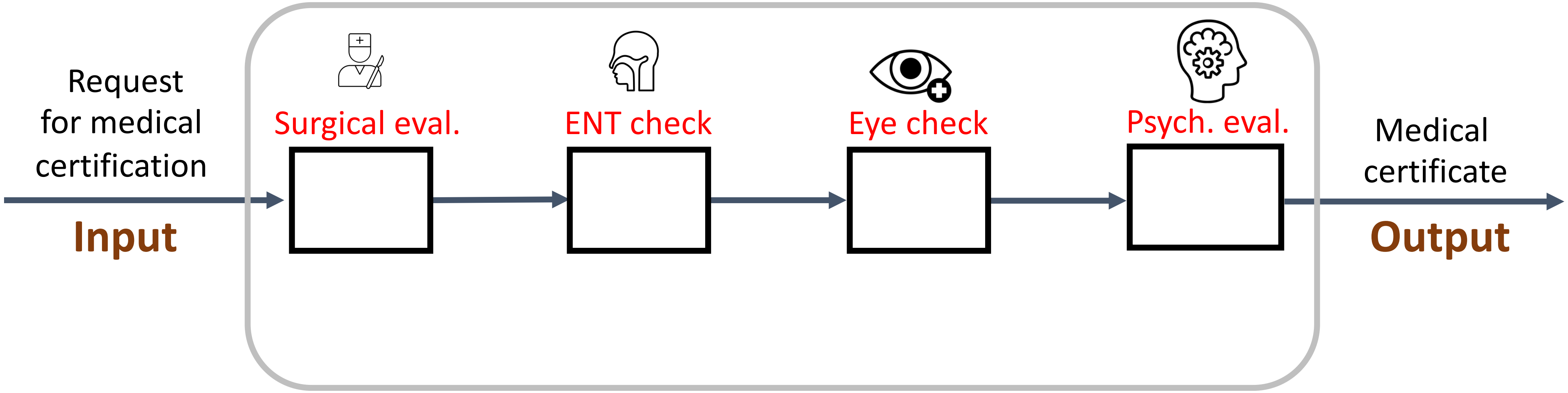
3. Improvement decisions
   A. Recommend on how to improve the system

Example: Form 086/y

1. The process flow diagram:
   - Determine the inputs and outputs:

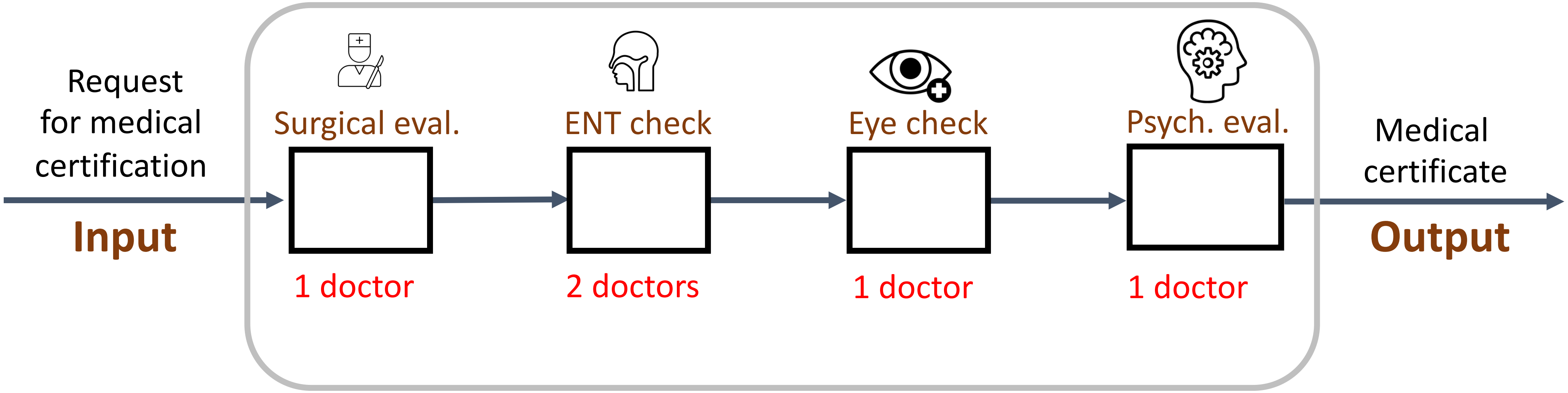
Example: Form 086/y

1. The process flow diagram:
   - Determine the tasks and their sequence:



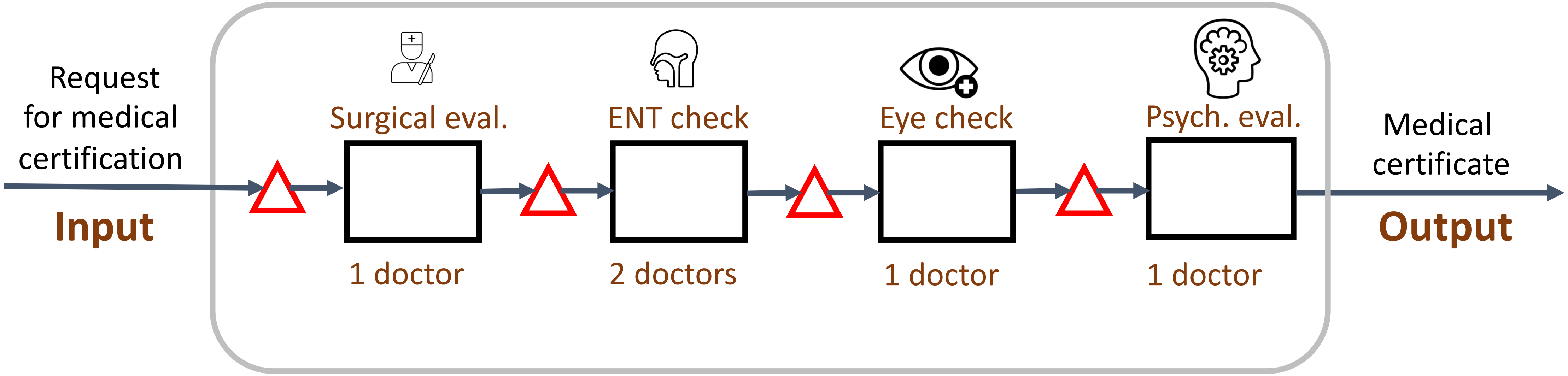
Example: Form 086/y

1. The process flow diagram:
   - Determine the resources used for each task:



Example: Form 086/y

1. The process flow diagram:
   - Determine where inventory is kept in the process:



Inventory: students.
Where is it kept? Waiting rooms, if you're lucky. Hallways, if you're not.

Definitions

- Inventory (work-in-process): how many units are inside the process
- Flow time: time spent by a unit within a process
- Cycle time: time between two successive units
- Throughput (flow rate): how many units flow through process in a unit of time
- Capacity: maximum possible throughput rate
- Utilization: throughput rate/capacity

Analyzing a process: a recipe

1. Write down the process flow diagram. Determine:
   A. The inputs and the outputs
   B. The tasks and their sequence
   C. Resources used for each task
   D. Where is inventory kept in the process

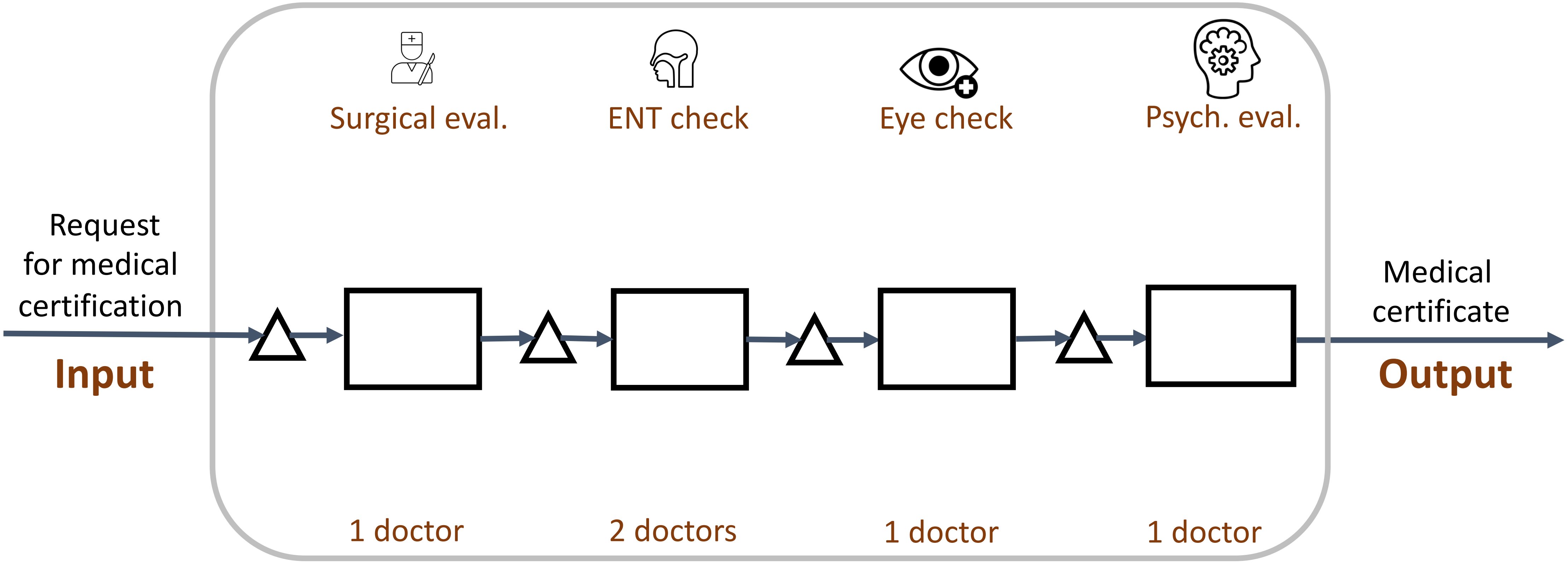
2. Bottleneck analysis. Determine:
   A. The capacity of each resource and the demand rate
   B. The bottleneck resource
   C. The capacity of the process

3. Improvement decisions
   A. Recommend on how to improve the system

Example: Form 086/y

2. Bottleneck analysis

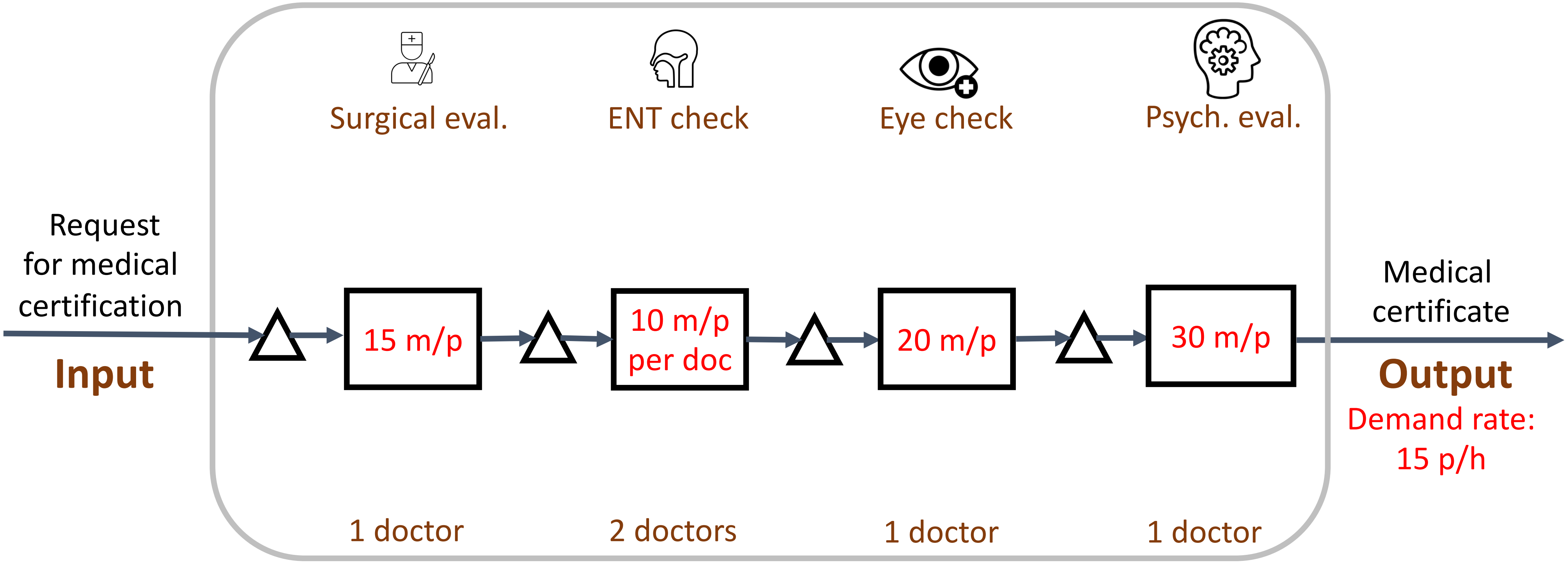
- Determine the capacity of each stage/resource:



Example: Form 086/y

2. Bottleneck analysis

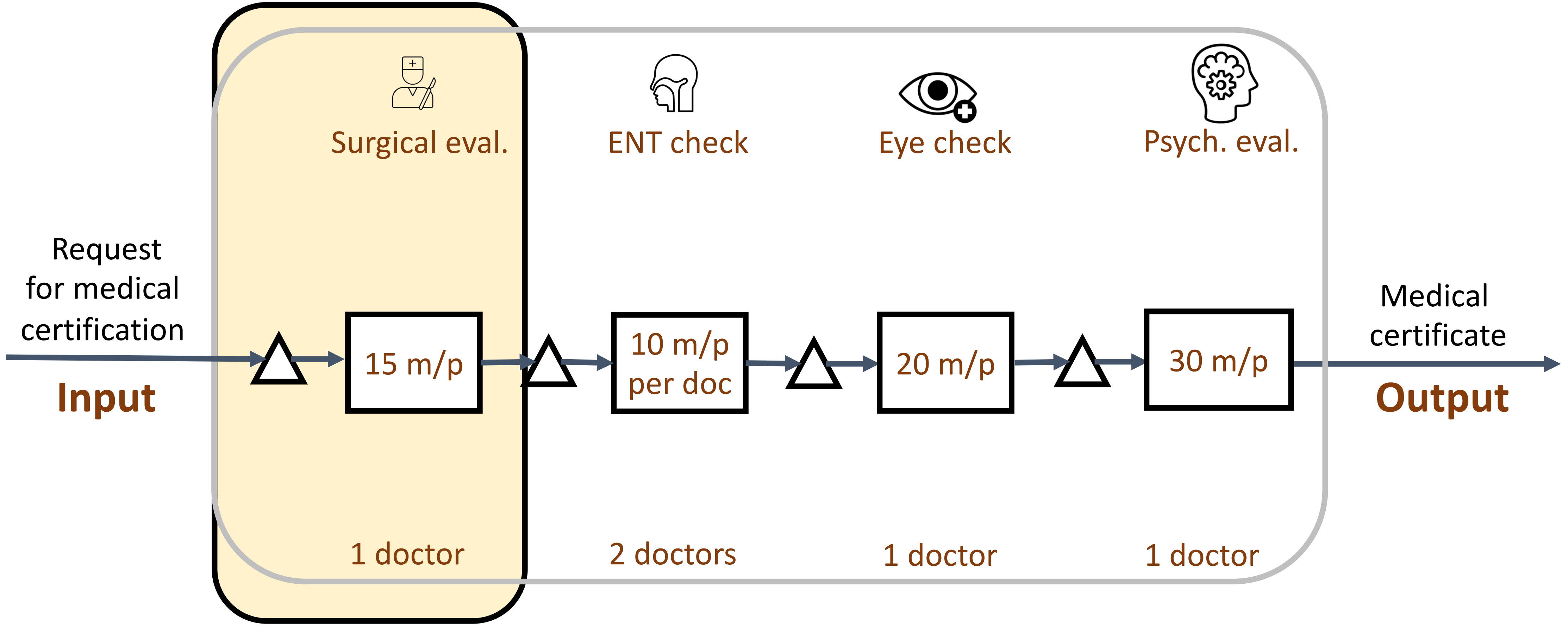
- Determine the capacity of each stage/resource:



Example: Form 086/y

2. Bottleneck analysis

- Determine the capacity of each stage/resource:



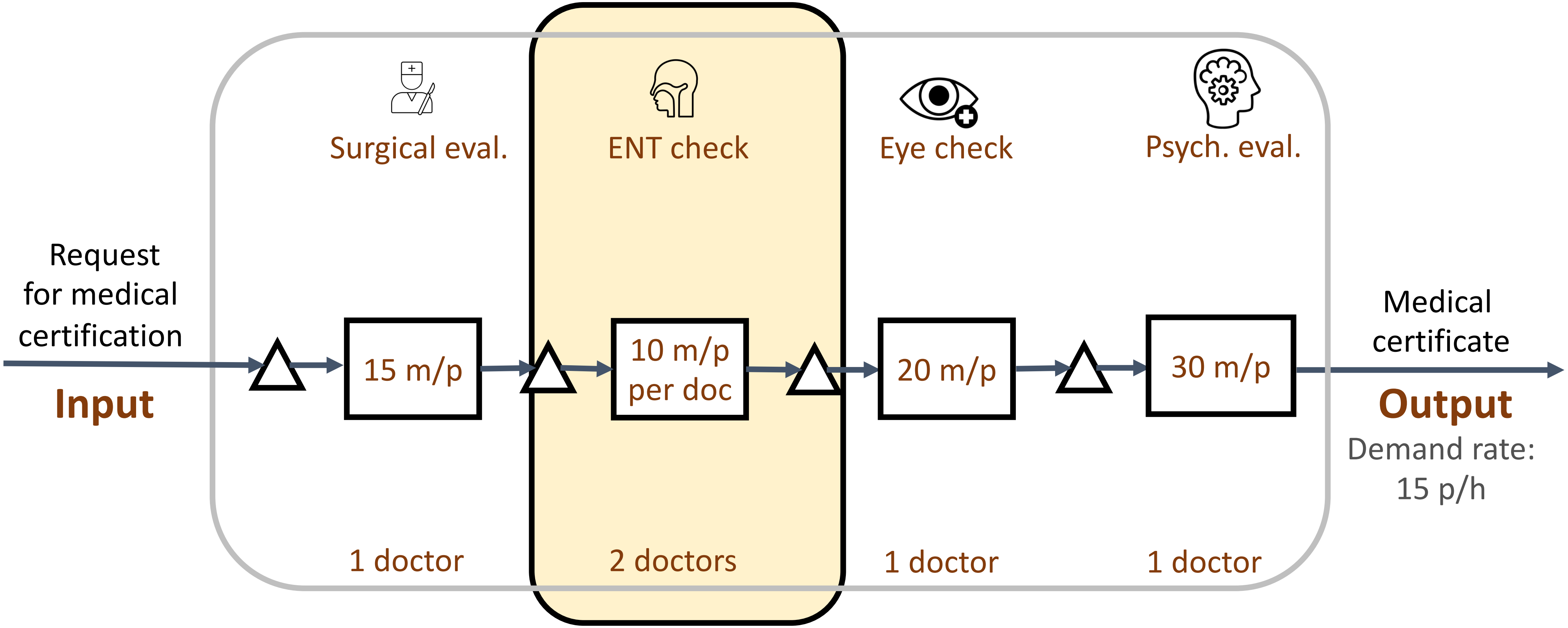
Surgical: 15 mins per person per doctor, 1 doctor

Cycle time: 15 mins. Flow time: 15 mins. Capacity: 60/15 = 4 persons/hour.

Example: Form 086/y

2. Bottleneck analysis

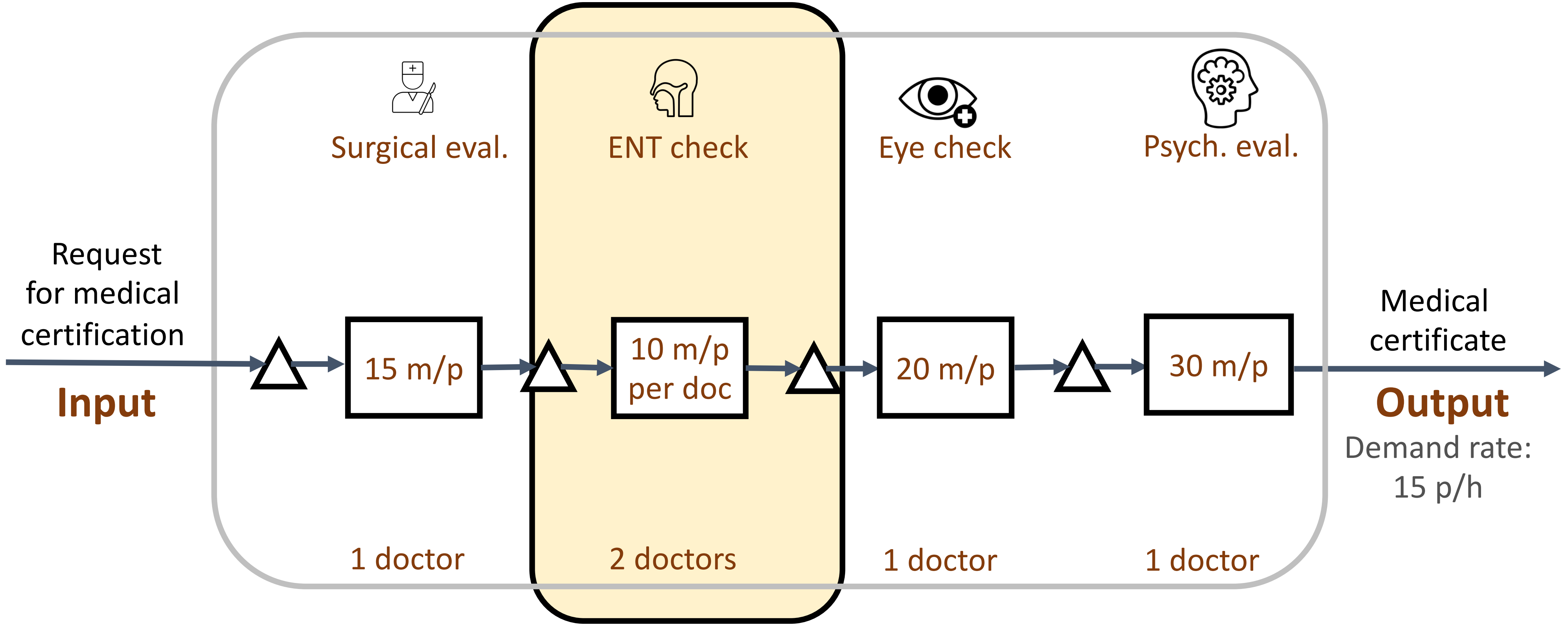
- Determine the capacity of each stage/resource:



Example: Form 086/y

2. Bottleneck analysis

- Determine the capacity of each stage/resource:



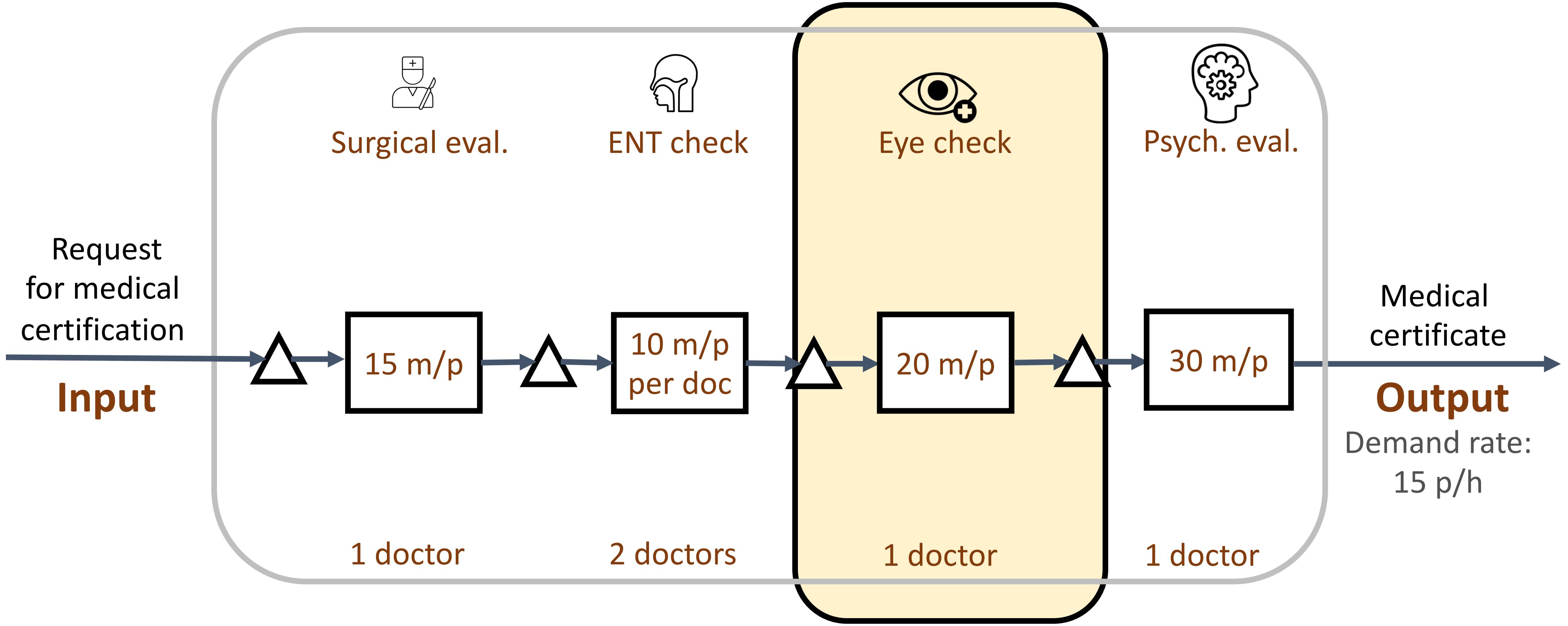
ENT: 10 mins per person per doctor, 2 doctors

Cycle time: 5 mins. Flow time: 10 mins. Capacity: 60/5 = 12 persons/hour.

Example: Form 086/y

2. Bottleneck analysis

- Determine the capacity of each stage/resource:



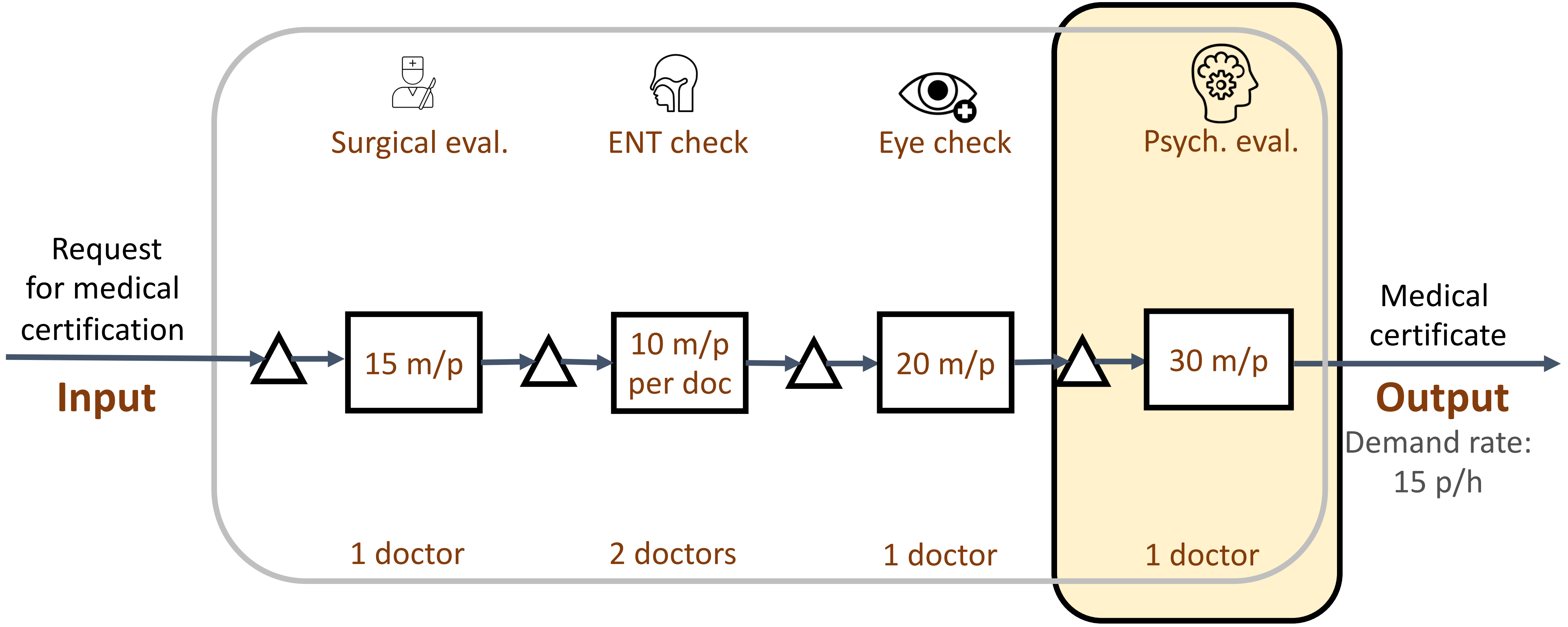
Eye check: 20 mins per person per doctor, 1 doctor

Cycle time: 20 mins. Flow time: 20 mins. Capacity: 60/20 = 3 persons/hour.

Example: Form 086/y

2. Bottleneck analysis

- Determine the capacity of each stage/resource:



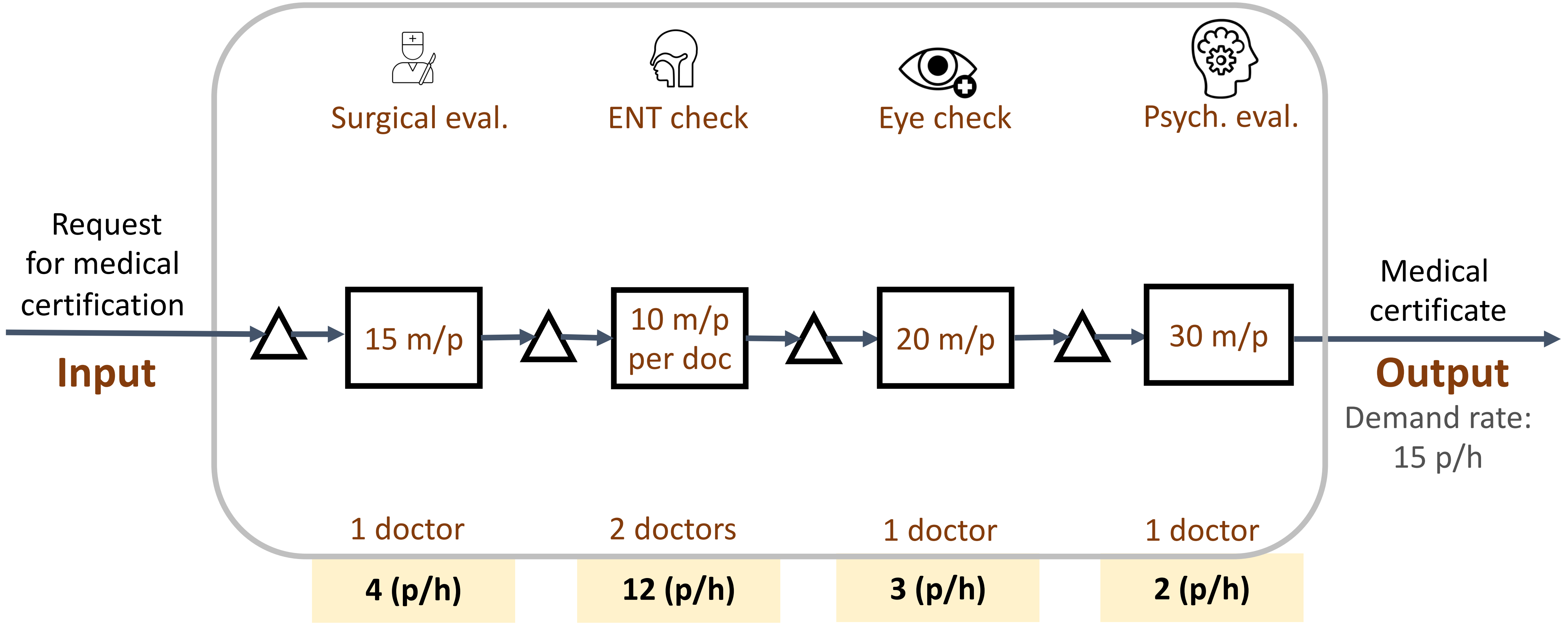
Psych. eval: 30 mins per person per doctor, 1 doctor

Cycle time: 30 mins. Flow time: 30 mins. Capacity: 60/30 = 2 persons/hour.

Example: Form 086/y

2. Bottleneck analysis

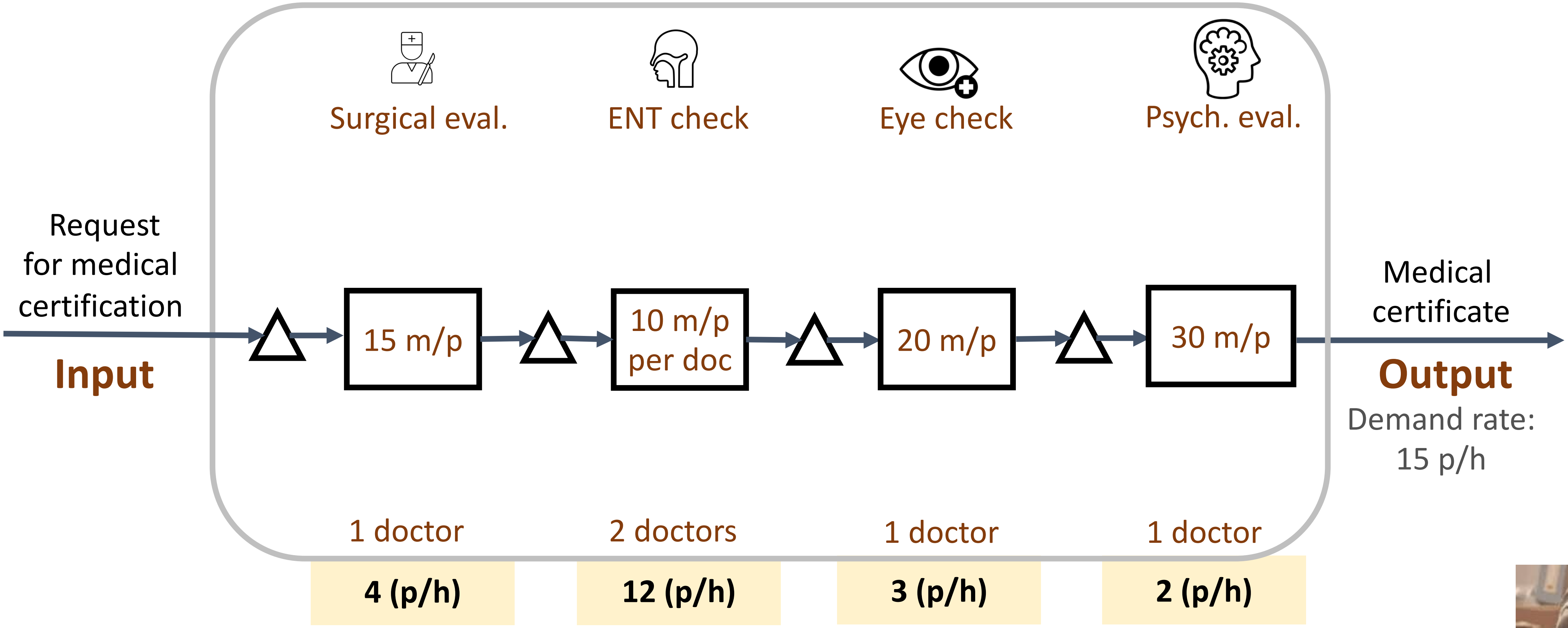
- The slowest stage is the bottleneck!



Example: Form 086/y

2. Bottleneck analysis

- The slowest stage is the bottleneck!

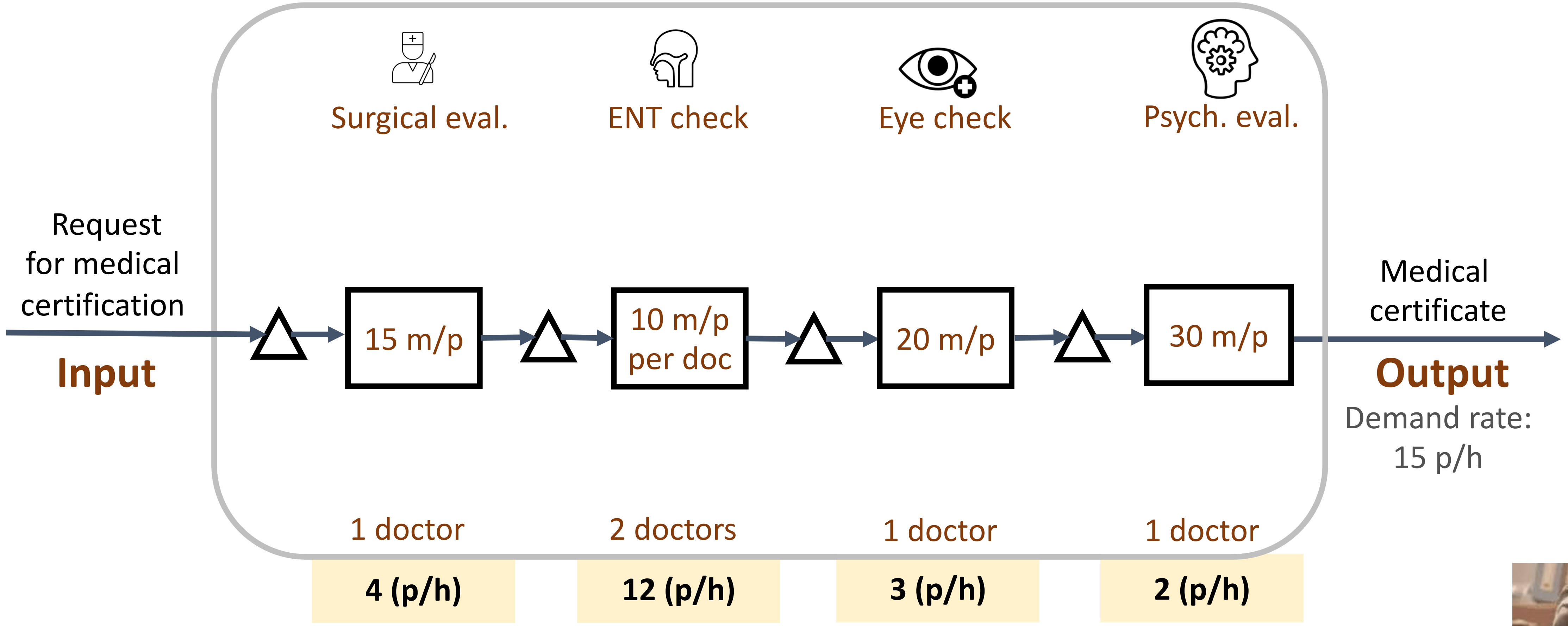


Psych. eval is the bottleneck! The process capacity is 2(p/h)

Example: Form 086/y

2. Bottleneck analysis

- The slowest stage is the bottleneck!

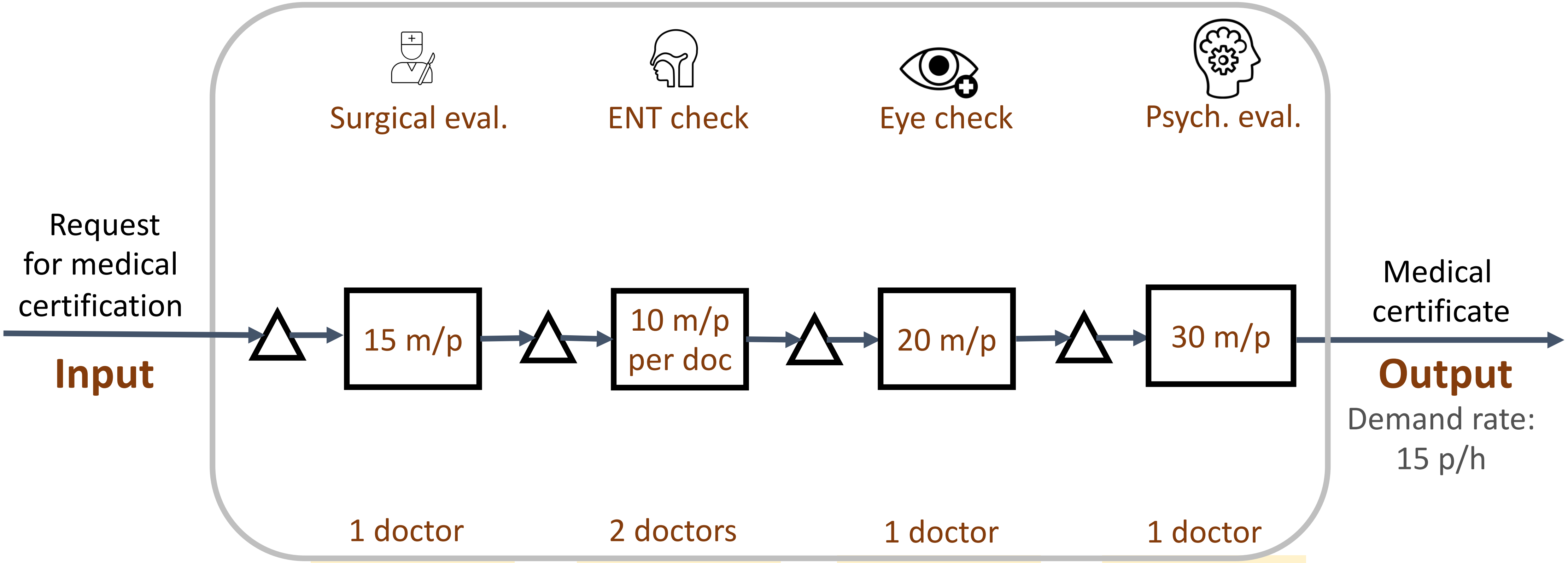


In general, the capacity of the process is equal to the capacity of the bottleneck

Example: Form 086/y

2. Bottleneck analysis

- The slowest stage is the bottleneck!



	Surgical eval.	ENT check	Eye check	Psych. eval.
	4 (p/h)	**12 (p/h)**	**3 (p/h)**	**2 (p/h)**
Utilizations (if running at full speed)	**50%**	**17%**	**66%**	**100%**

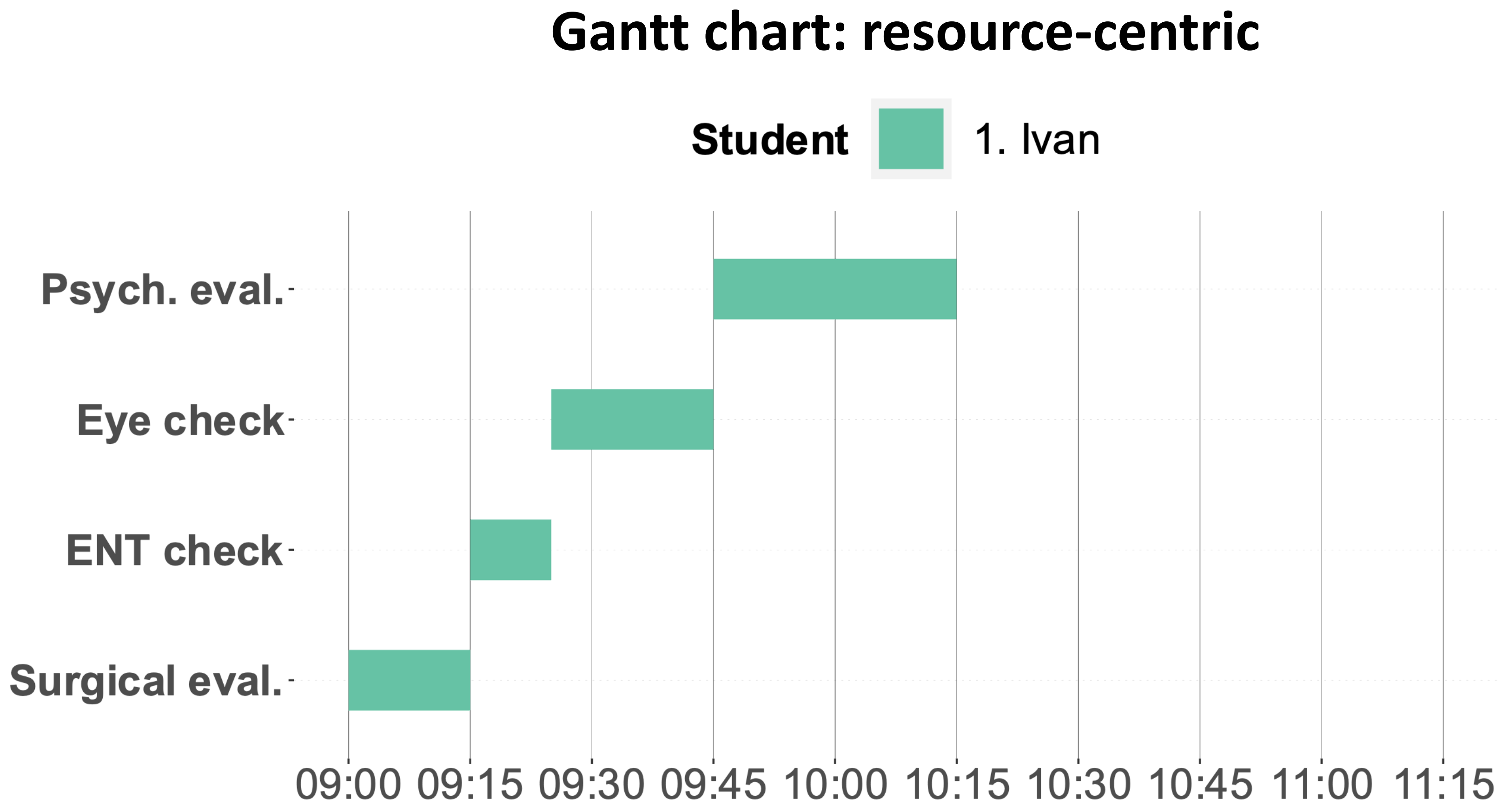
Helpful visualizations

Gantt chart: resource-centric

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

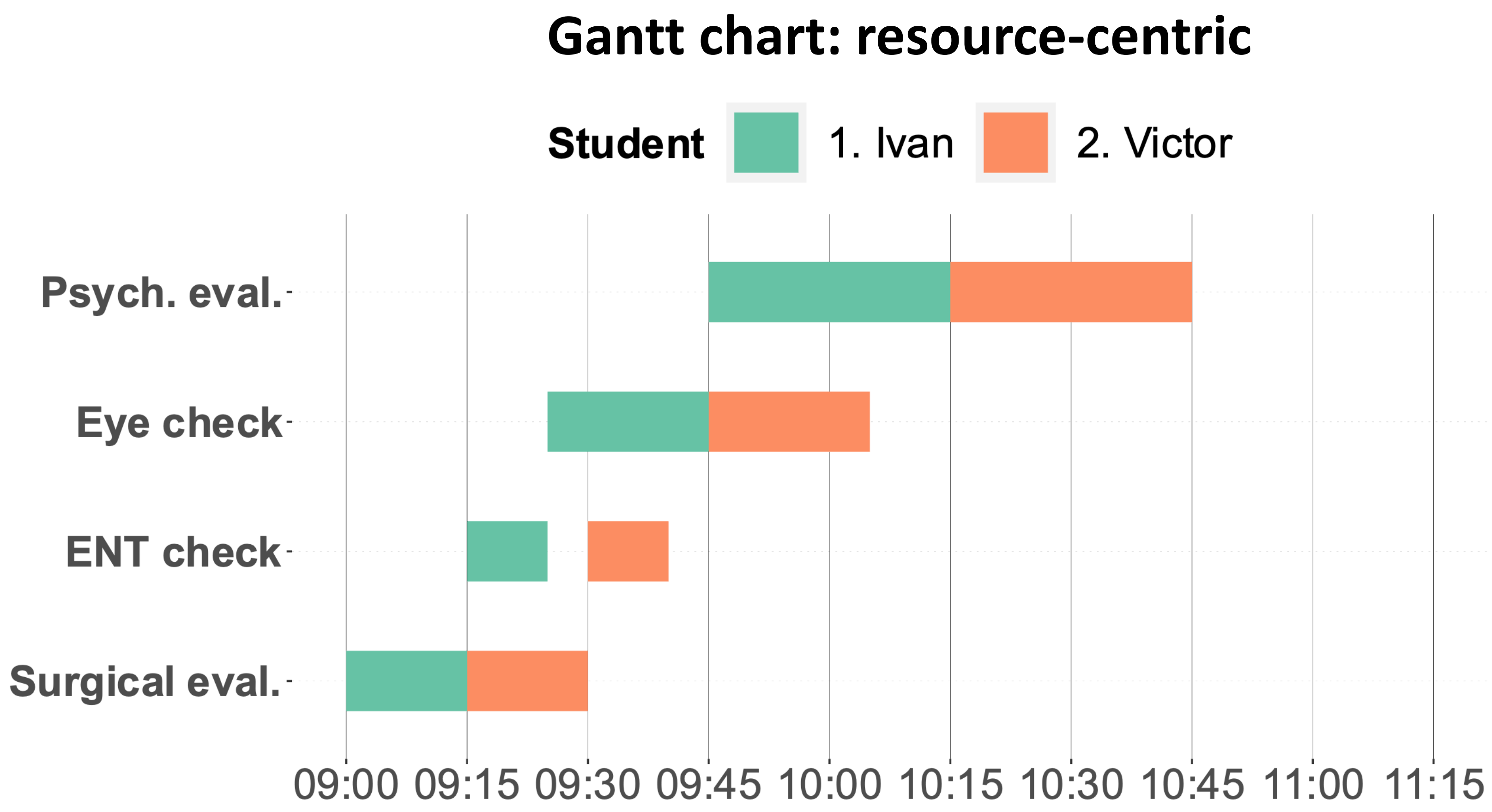
Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min



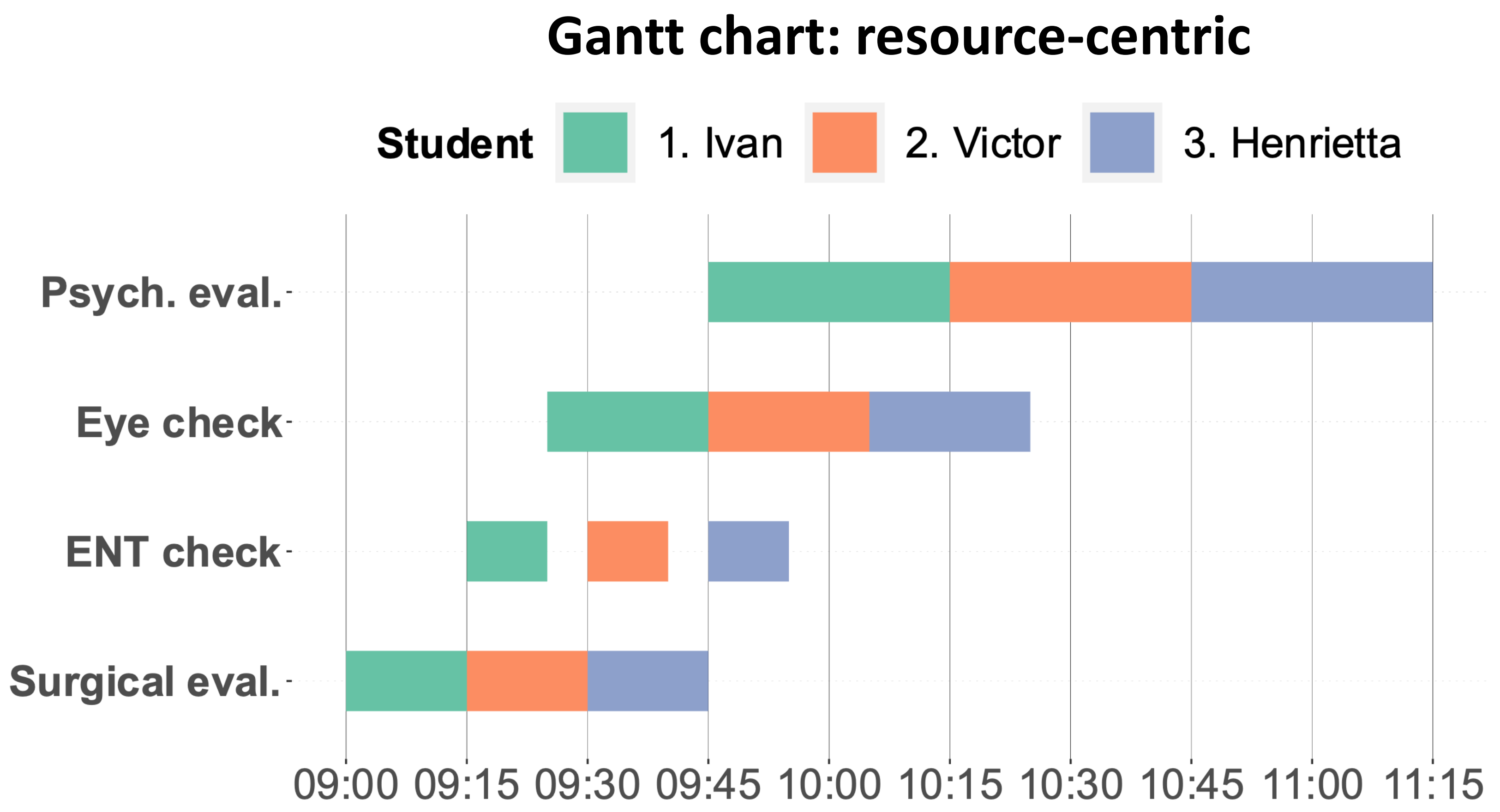
Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min



Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min



Helpful visualizations

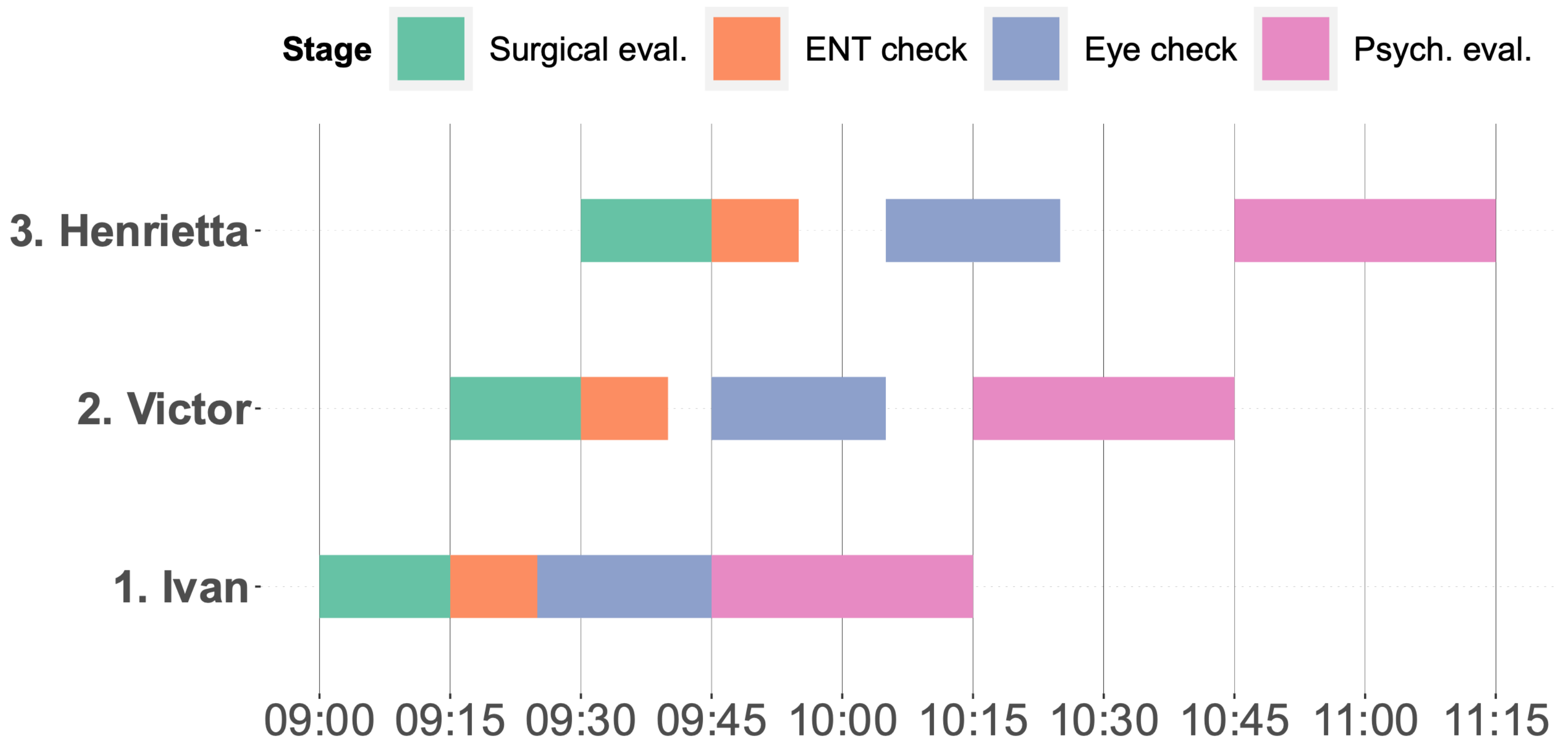
Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

Gantt chart: unit-centric

Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

Gantt chart: unit-centric



Helpful visualizations

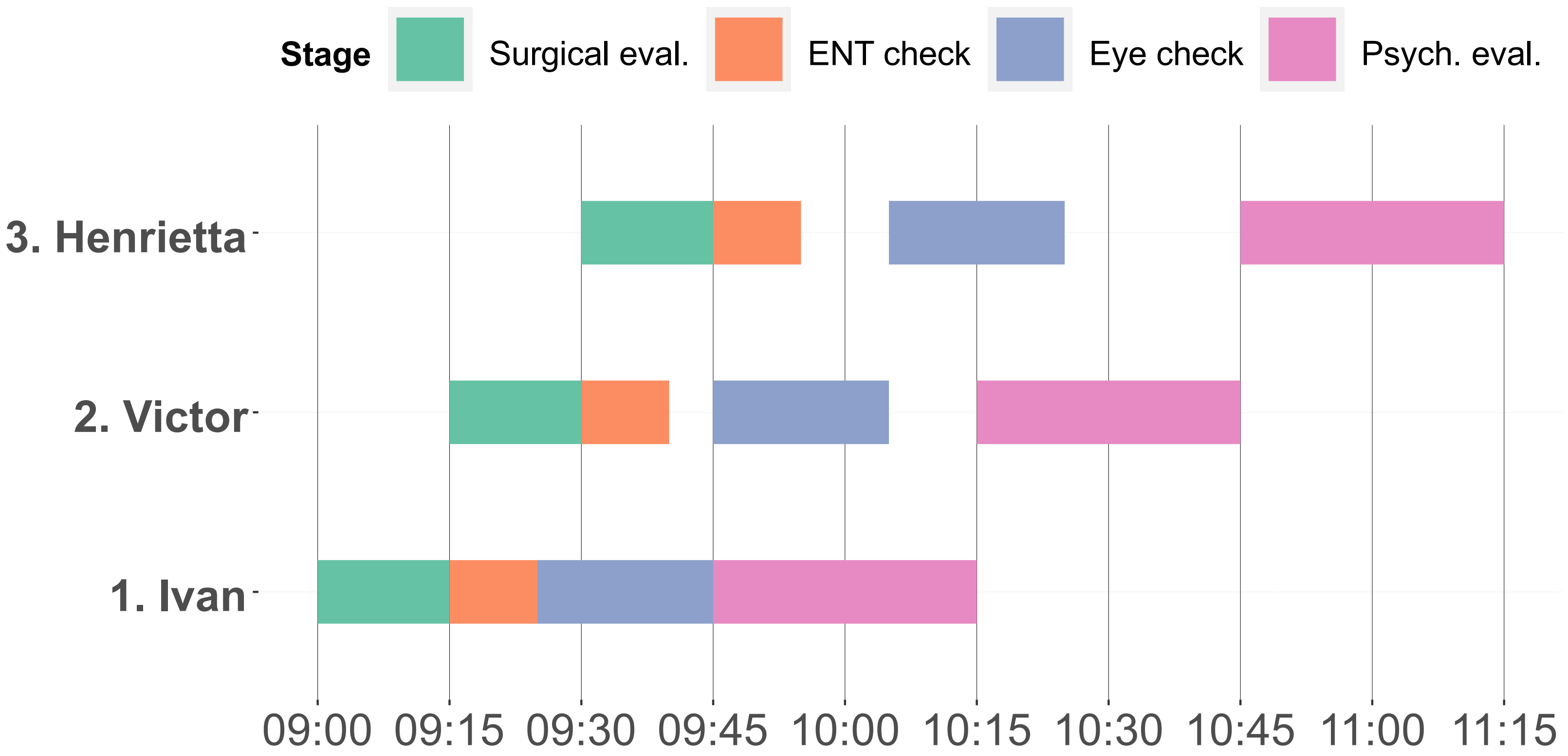
Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

Inventory buildup diagram

Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

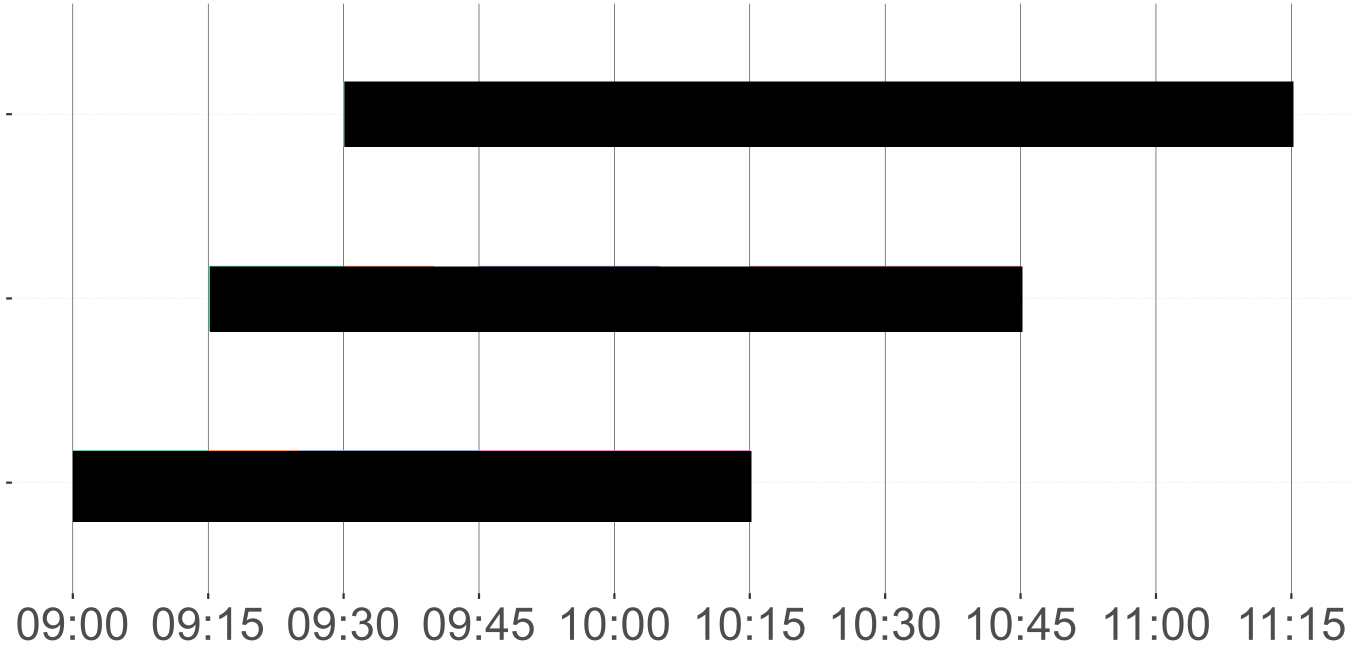
Inventory buildup diagram



Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

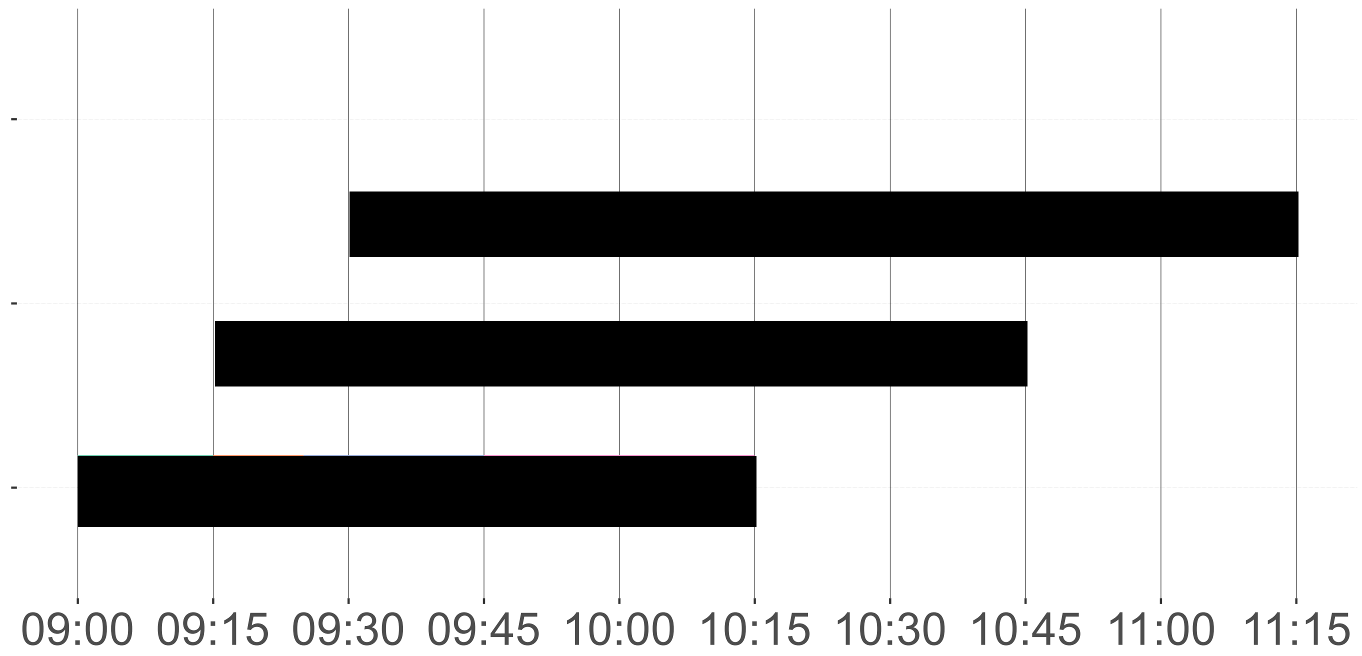
Inventory buildup diagram



Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

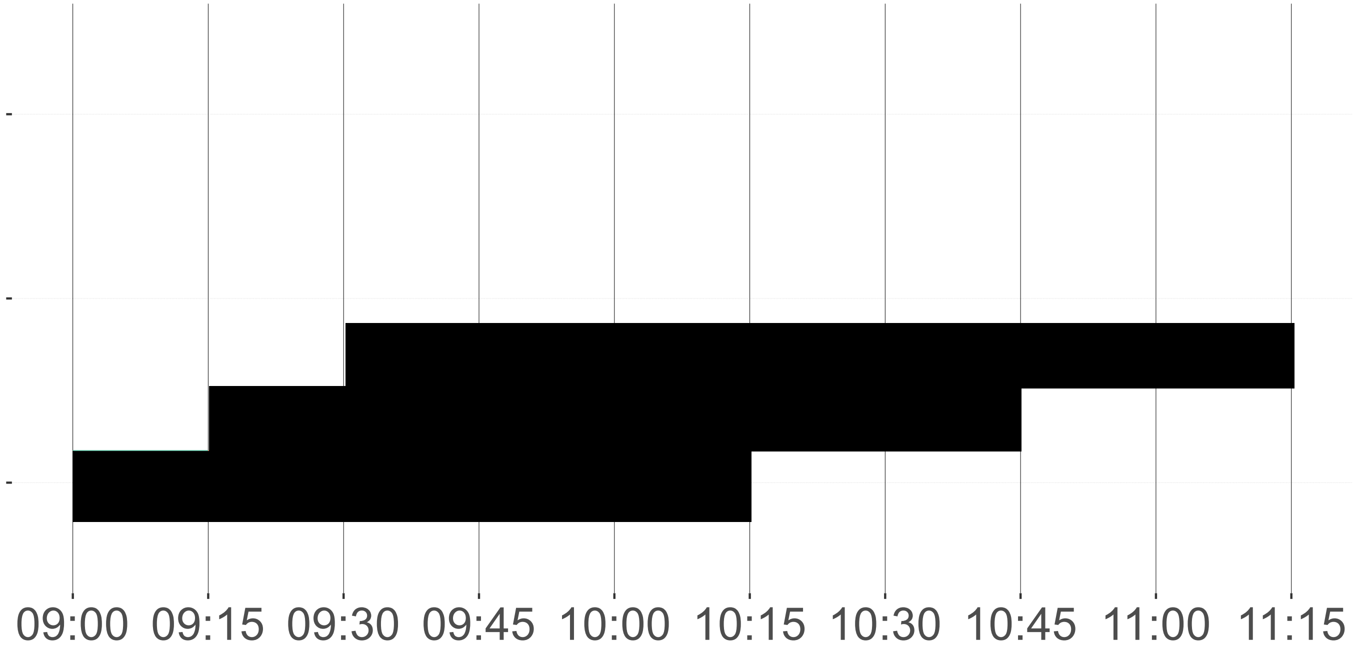
Inventory buildup diagram



Helpful visualizations

Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

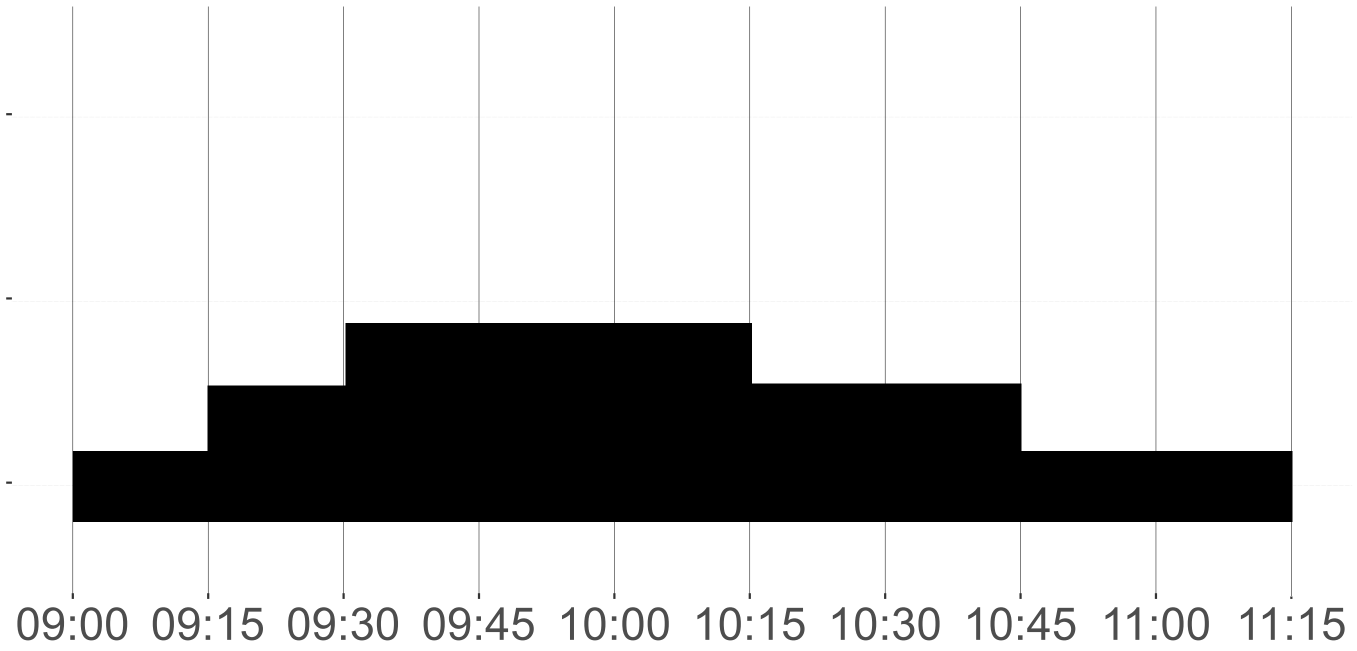
Inventory buildup diagram



Helpful visualizations

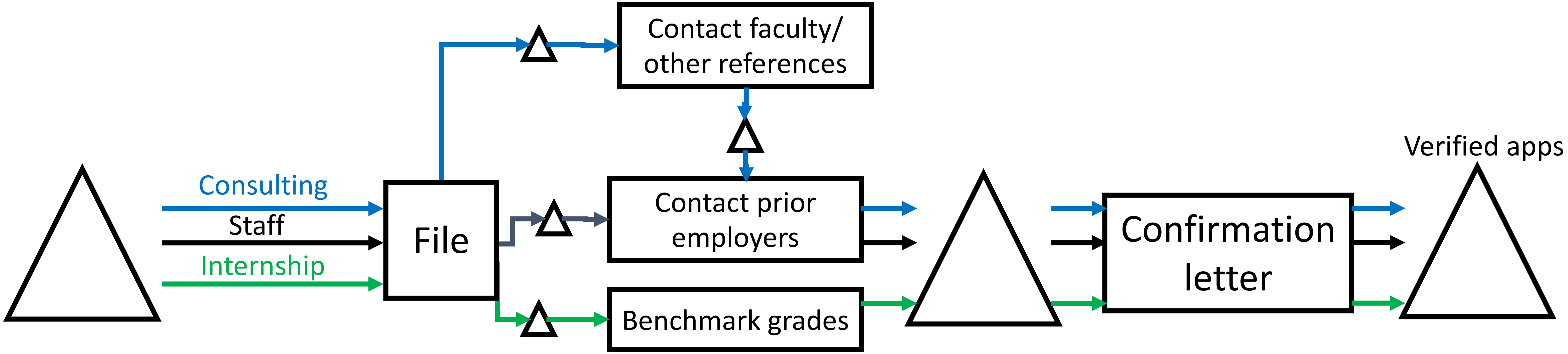
Student	Stage	Start	Duration
1. Ivan	Surgical eval.	9:00	15 min
1. Ivan	ENT check	9:15	10 min
1. Ivan	Eye check	9:25	20 min
1. Ivan	Psych. eval.	9:45	30 min
2. Victor	Surgical eval.	9:15	15 min
2. Victor	ENT check	9:30	10 min
2. Victor	Eye check	9:45	20 min
2. Victor	Psych. eval.	10:15	30 min
3. Henrietta	Surgical eval.	9:30	15 min
3. Henrietta	ENT check	9:45	10 min
3. Henrietta	Eye check	10:05	20 min
3. Henrietta	Psych. eval.	10:35	30 min

Inventory buildup diagram



Multiple flow units

Multiple flow units: example



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

	Proc. time	**# wrk.**
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

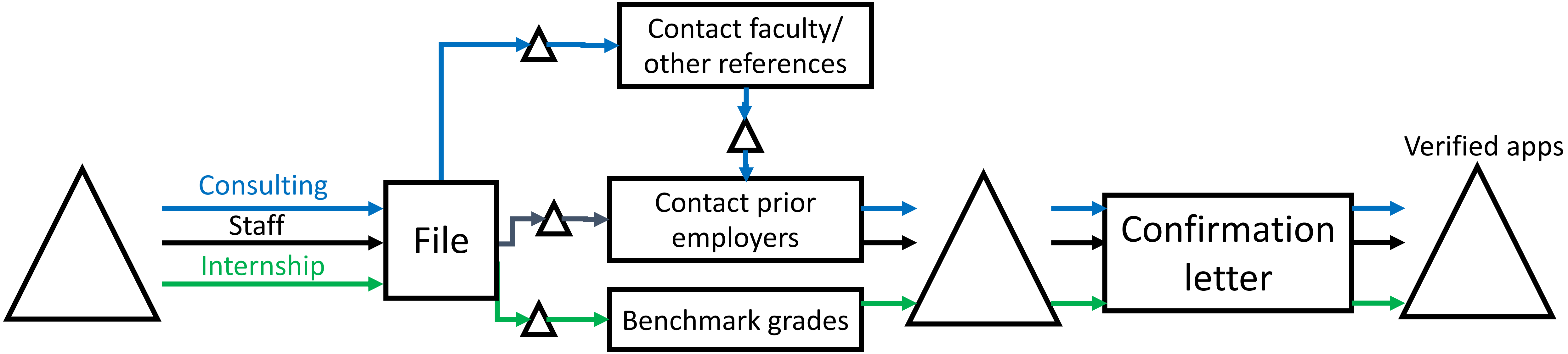
We have multiple types of applications.

A bottleneck is not always the slowest stage.

Instead, it is the stage with the highest implied utilization.

Implied utilization of a stage is equal to $\frac{\text{Demand}}{\text{Capacity}}$

Multiple flow units: example

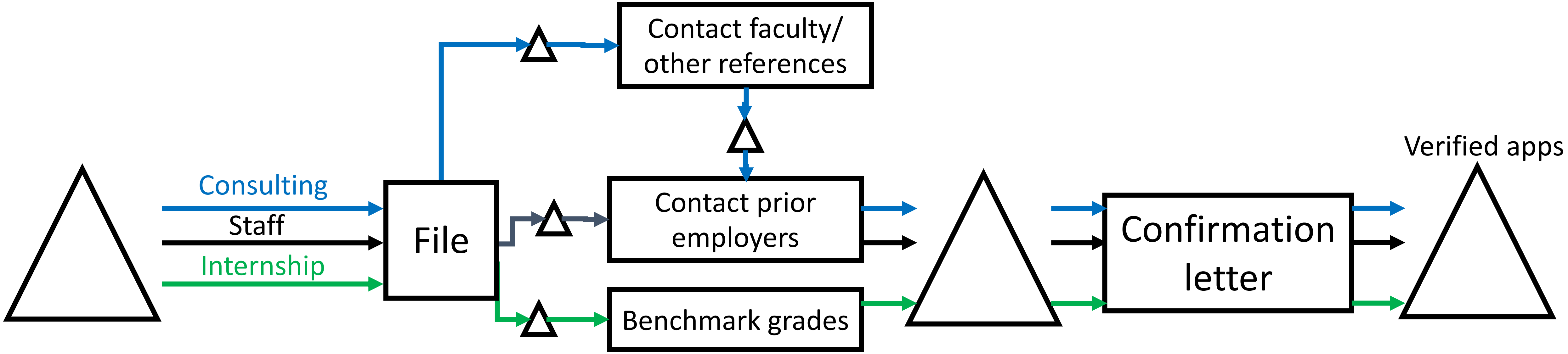


Demand: 3 consulting, 11 staff, and 4 internship applications per hour

Approach 1: flow unit is an **application**

				Workload (apps/hour)				
	Proc. time	**# wrk.**	**Capacity**	**Consult.**	**Staff**	**Interns**	**Total**	**Impl. Util.**
File	3 [min/app]	1	20 [app/h]					
C/ faculty	20 [min/app]	2	6 [app/h]					
C/ employers	15 [min/app]	3	12 [app/h]					
Grade bench.	8 [min/app]	2	15 [app/h]					
Confirmation	2 [min/app]	1	30 [app/h]					

Multiple flow units: example

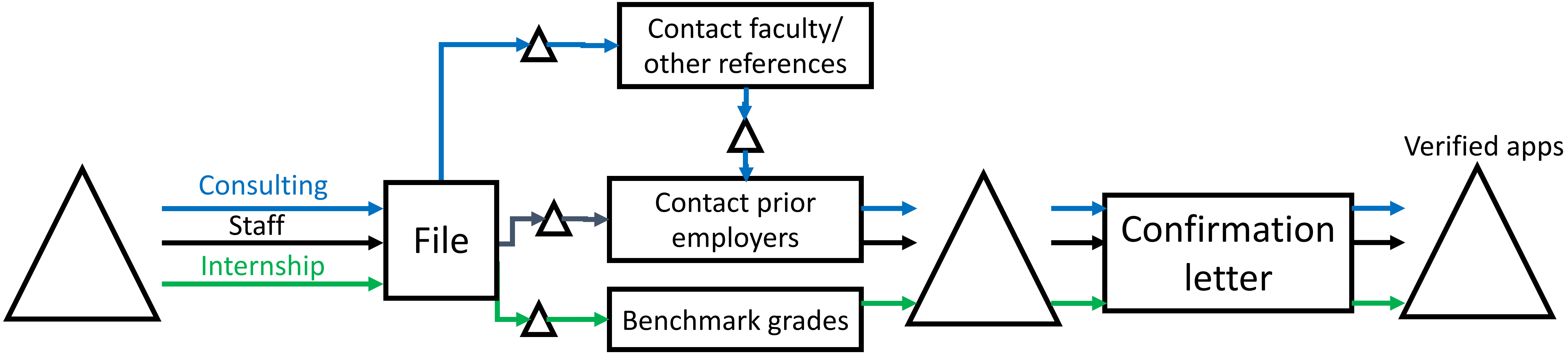


Demand: 3 consulting, 11 staff, and 4 internship applications per hour

Approach 1: flow unit is an **application**

				Workload (apps/hour)				
	Proc. time	**# wrk.**	**Capacity**	**Consult.**	**Staff**	**Interns**	**Total**	**Impl. Util.**
File	3 [min/app]	1	20 [app/h]	3	11	4	18	
C/ faculty	20 [min/app]	2	6 [app/h]	3	0	0	3	
C/ employers	15 [min/app]	3	12 [app/h]	3	11	0	14	
Grade bench.	8 [min/app]	2	15 [app/h]	0	0	4	4	
Confirmation	2 [min/app]	1	30 [app/h]	3	11	4	18	

Multiple flow units: example



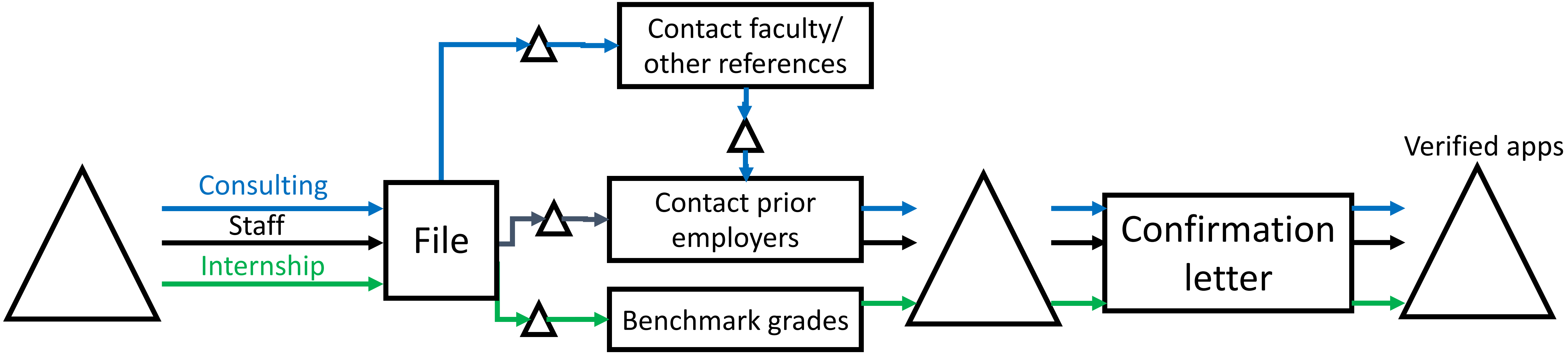
Demand: 3 consulting, 11 staff, and 4 internship applications per hour

Approach 1: flow unit is an **application**

				Workload (apps/hour)				
	Proc. time	**# wrk.**	**Capacity**	**Consult.**	**Staff**	**Interns**	**Total**	**Impl. Util.**
File	3 [min/app]	1	20 [app/h]	3	11	4	18	90%
C/ faculty	20 [min/app]	2	6 [app/h]	3	0	0	3	50%
C/ employers	15 [min/app]	3	12 [app/h]	3	11	0	14	117%
Grade bench.	8 [min/app]	2	15 [app/h]	0	0	4	4	27%
Confirmation	2 [min/app]	1	30 [app/h]	3	11	4	18	60%

Bottleneck

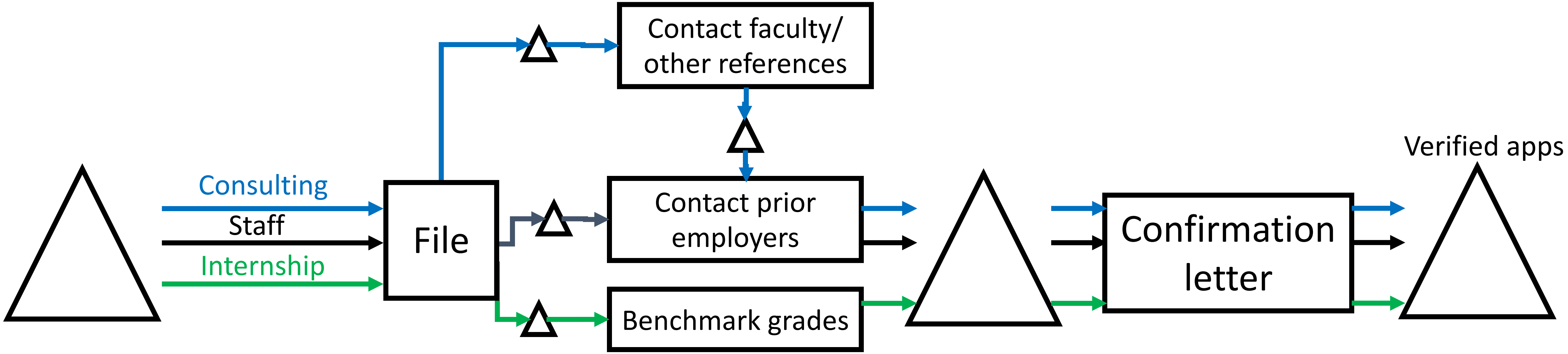
Multiple flow units: example



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

Approach 2: flow unit is **one minute of work**

Multiple flow units: example



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

Approach 2: flow unit is **one minute of work**

				Workload (min/hour)				
	Proc. time	**# wrk.**	**Capacity**	**Consult.**	**Staff**	**Interns**	**Total**	**Impl. Util.**
File	3 [min/app]	1	60 [min/h]	3 × 3	3 × 11	4 × 3	54	90%
C/ faculty	20 [min/app]	2	120 [min/h]	3 × 20	0	0	60	50%
C/ employers	15 [min/app]	3	180 [min/h]	3 × 15	11 × 15	0	210	117%
Grade bench.	8 [min/app]	2	120 [min/h]	0	0	4 × 8	32	27%
Confirmation	2 [min/app]	1	60 [app/h]	3 × 2	11 × 2	4 × 2	36	60%

Why use this approach?

Suppose different app. types take different times to file.

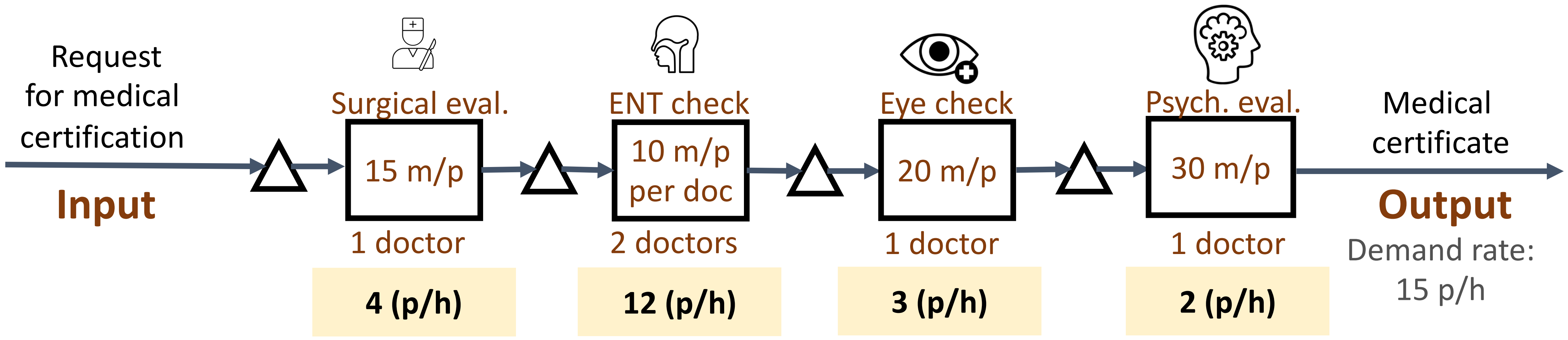
Up next

- Can we solve process analysis problems automatically?
- Can we consider more complicated process types?
- Answer: Linear programming
- Can we then apply the methodology to other settings?
- How to implement in code: Python, numpy, cvxpy

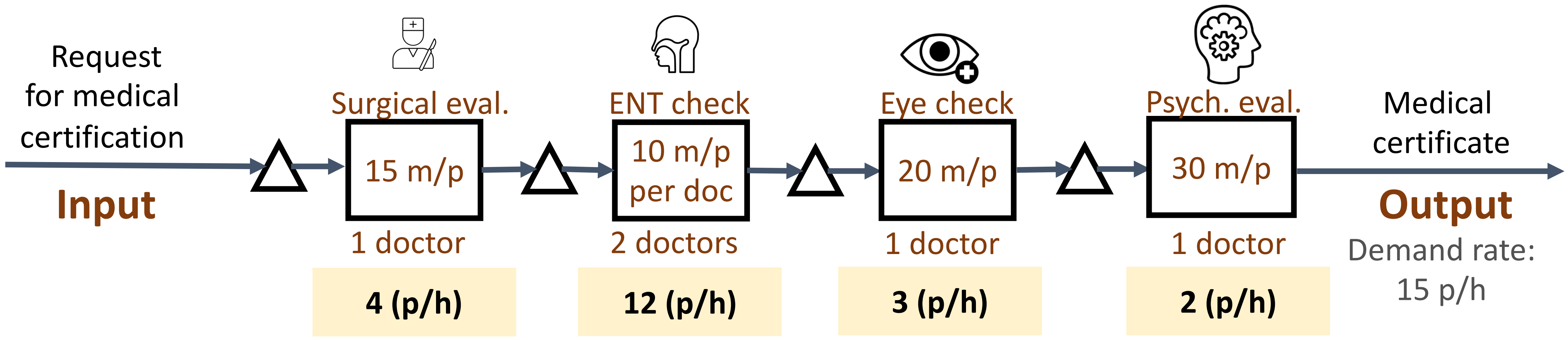
Business analytics I
Operations Analytics
Class 2
Linear Programming Formulations

Marat Salikhov
March 14th, 2022

Recap: process analysis



Recap: process analysis



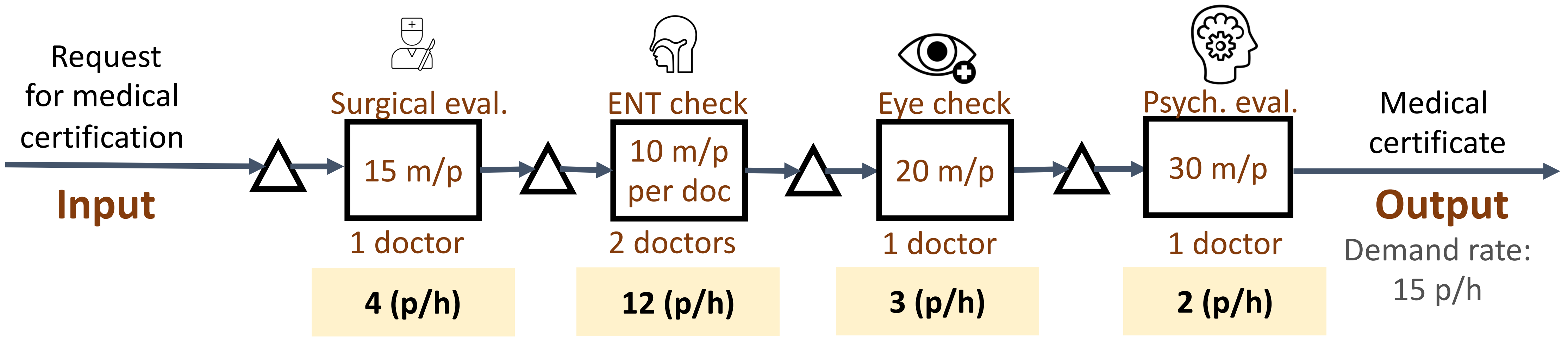
Variable **t**: throughput

Objective: maximize throughput

Constraints:

- $t \leq 4$ (Surgeon)
- $t \leq 12$ (ENT)
- $t \leq 3$ (eye check)
- $t \leq 2$ (psych)

Recap: process analysis



Variable **t**: throughput

Objective: maximize throughput

Constraints:

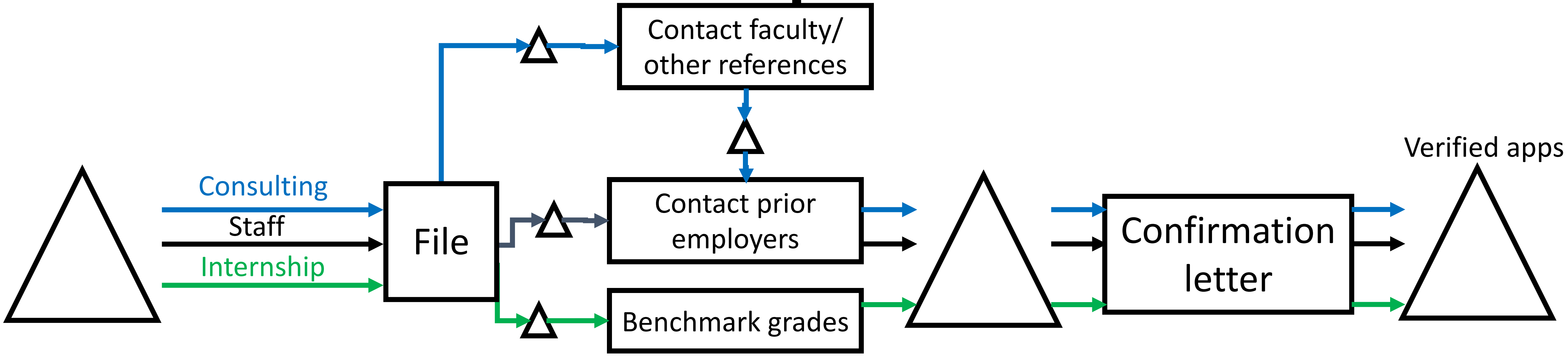
- $t \leq 4$ (Surgeon)
- $t \leq 12$ (ENT)
- $t \leq 3$ (eye check)
- $t \leq 2$ (psych)

Solution: t = 2

In general, throughput is equal to the capacity of the slowest stage

The constraint that is satisfied with equality represents the bottleneck

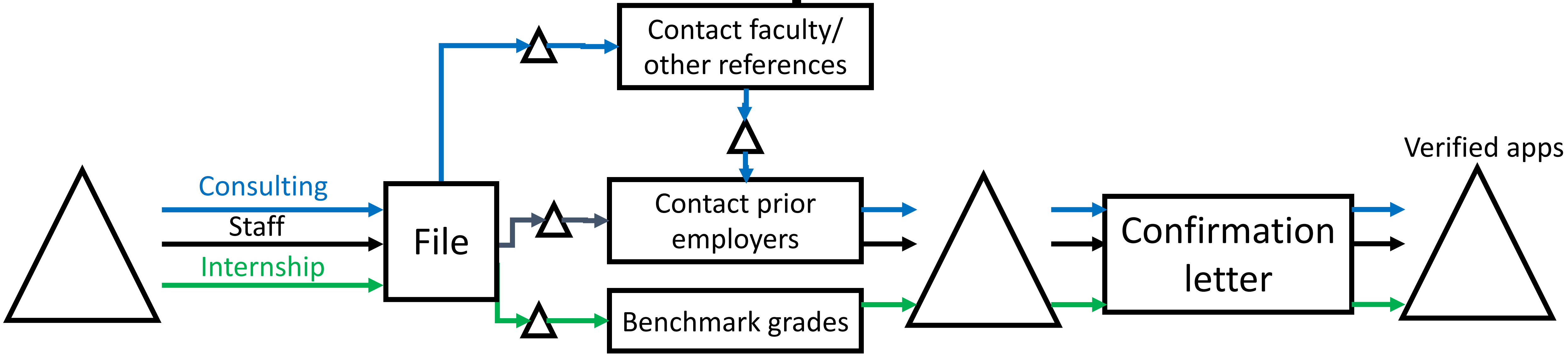
A more complicated one



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

[redacted]		
	Proc. time	**# wrk.**
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

A more complicated one

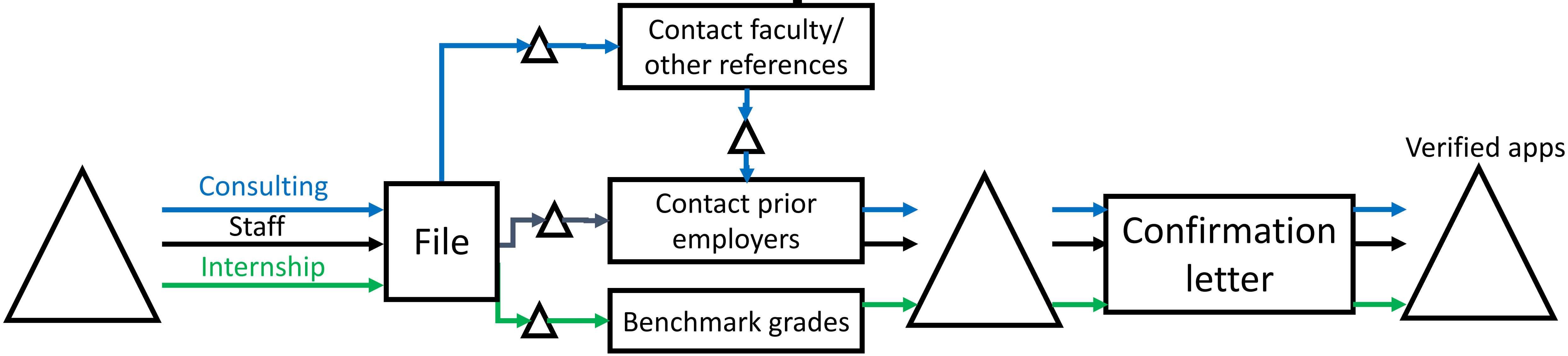


Demand: 3 consulting, 11 staff, and 4 internship applications per hour

	Proc. time	# wrk.
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

Variables: **throughput** for C, S, and I

A more complicated one



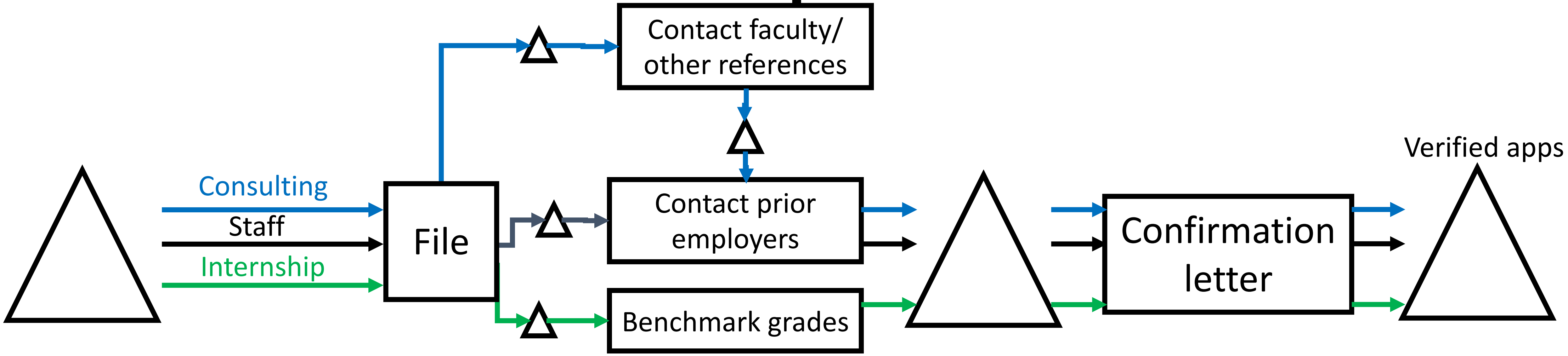
Demand: 3 consulting, 11 staff, and 4 internship applications per hour

	Proc. time	# wrk.
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

Variables: **throughput** for C, S, and I

Objective: maximize C + S + I

A more complicated one



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

	Proc. time	**# wrk.**
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

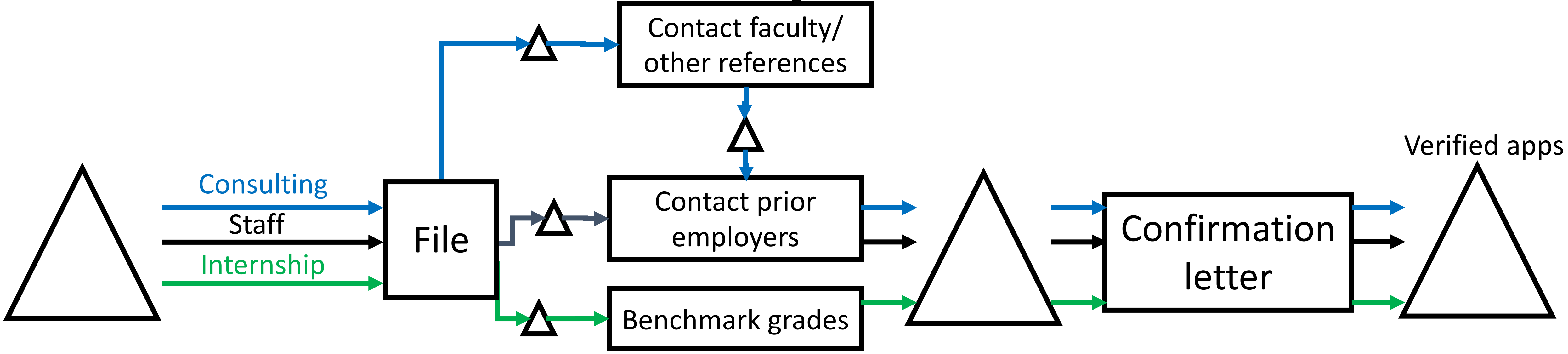
Variables: **throughput** for C, S, and I

Objective: maximize C + S + I

Constraints

- $3(C + S + I) \leq 60$ (filing step)
- $20C \leq 120$ (contact faculty)
- $15(C + S) \leq 180$ (contact employers)
- $8I \leq 120$ (benchmark grades)
- $2(C + S + I) \leq 60$ (confirmation)

A more complicated one



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

	Proc. time	**# wrk.**
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

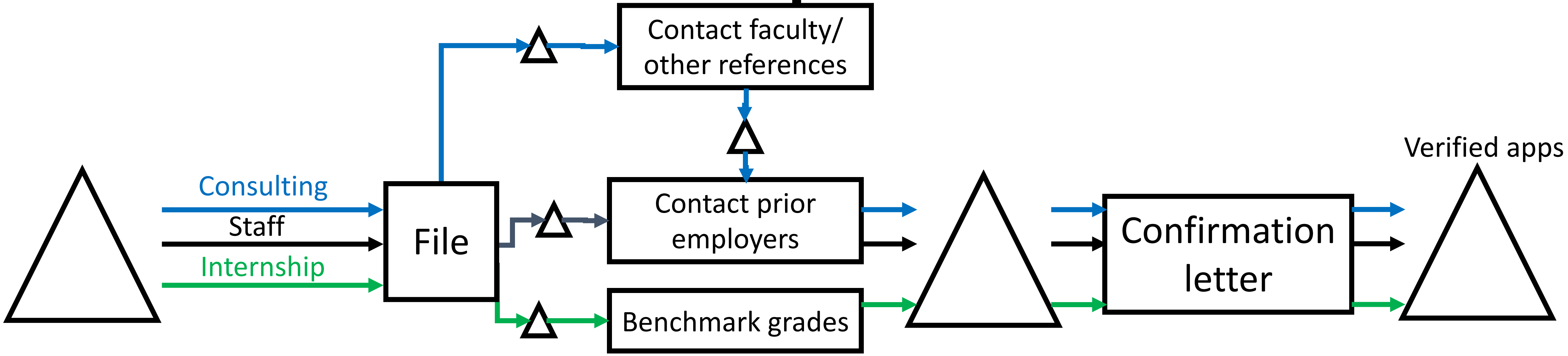
Variables: **throughput** for C, S, and I

Objective: maximize C + S + I

Constraints

- $3(C + S + I) \leq 60$ (filing step)
- $20C \leq 120$ (contact faculty)
- $15(C + S) \leq 180$ (contact employers)
- $8I \leq 120$ (benchmark grades)
- $2(C + S + I) \leq 60$ (confirmation)
- $0 \leq C \leq 3, 0 \leq S \leq 11, 0 \leq I \leq 4$ (demand)
- $C/3 = S/11 = I/4$ (fairness)

A more complicated one



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

	Proc. time	**# wrk.**
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

Variables: **throughput** for C, S, and I

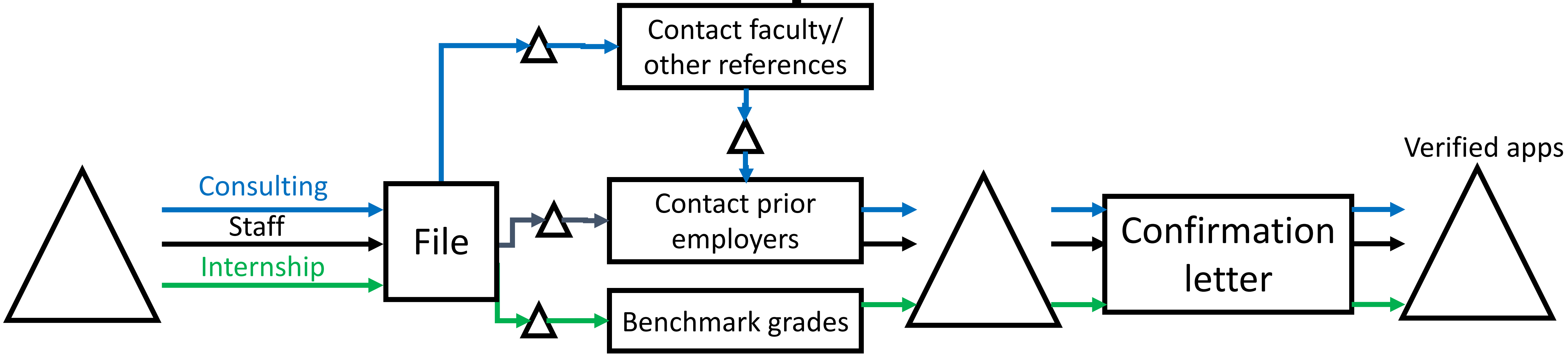
Objective: maximize C + S + I

Constraints

- $3(C + S + I) \leq 60$ (filing step)
- $20C \leq 120$ (contact faculty)
- $15(C + S) \leq 180$ (contact employers)
- $8I \leq 120$ (benchmark grades)
- $2(C + S + I) \leq 60$ (confirmation)
- $0 \leq C \leq 3, 0 \leq S \leq 11, 0 \leq I \leq 4$ (demand)
- $C/3 = S/11 = I/4$ (fairness)

Solution: C = 2.57, S = 9.42, I = 3.42
Objective: 15.42

A more complicated one



Demand: 3 consulting, 11 staff, and 4 internship applications per hour

	Proc. time	**# wrk.**
File	3 [min/app]	1
C/ faculty	20 [min/app]	2
C/ employers	15 [min/app]	3
Grade bench.	8 [min/app]	2
Confirmation	2 [min/app]	1

Variables: **throughput** for C, S, and I

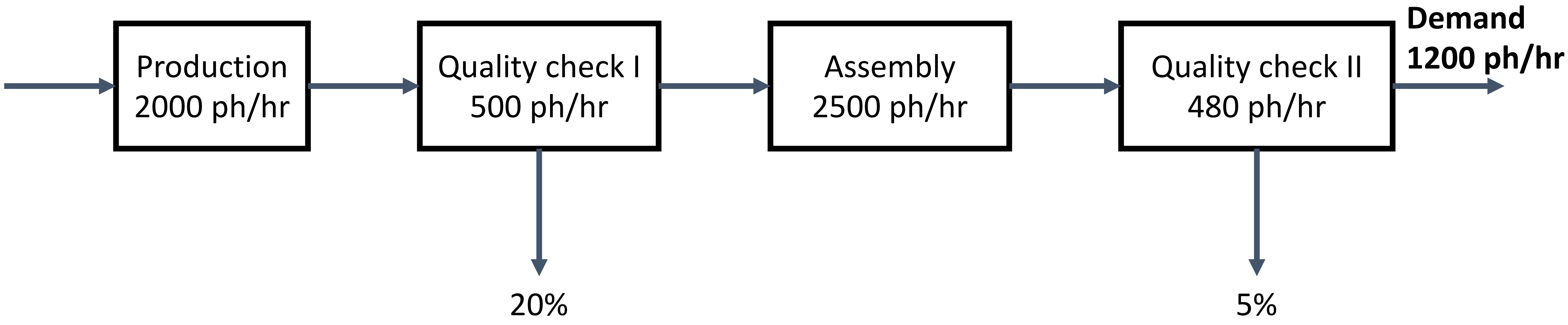
Objective: maximize C + S + I

Constraints

- $3(C+S+I) \leq 60$ (filing step)
- $20C \leq 120$ (contact faculty)
- $15(C+S) \leq 180$ (contact employers)
- $8I \leq 120$ (benchmark grades)
- $2(C+S+I) \leq 60$ (confirmation)
- $0 \leq C \leq 3, 0 \leq S \leq 11, 0 \leq I \leq 4$ (demand)
- $C/3 = S/11 = I/4$ (fairness)

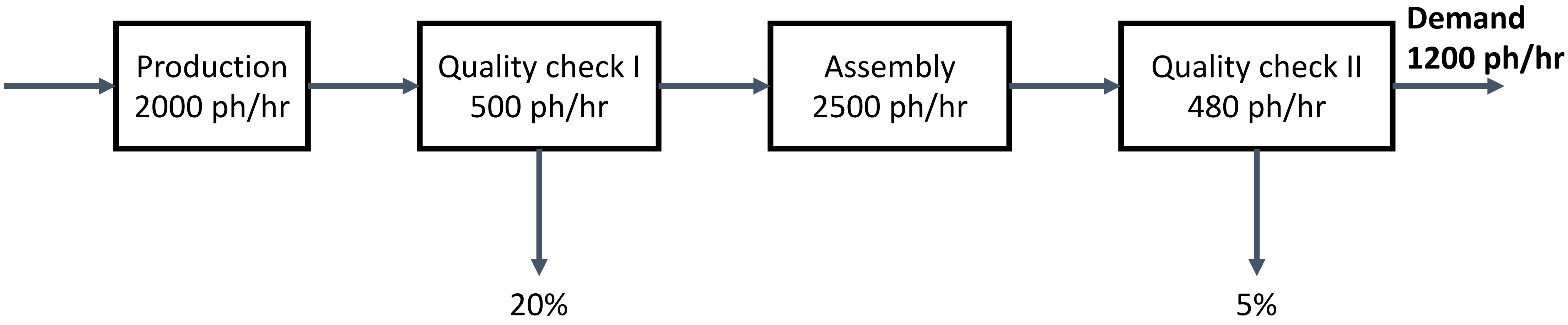
Solution: C = 2.57, S = 9.42, I = 3.42
Objective: 15.42

Try it yourself



Formulate the problem to find the capacity of this process

Try it yourself



Formulate the problem to find the capacity of this process

- Variable: t units of input at the start of the process
- Objective: maximize $0.95 \times 0.8 \times t$
- Constraints:
  - $t \leq 2000$ (Production)
  - $t \leq 500$ (Quality check I)
  - $0.8t \leq 2500$ (Assembly)
  - $0.8t \leq 480$ (Quality check II)
  - $0.8 \times 0.95 \times t \leq 1200$ (Demand)

Canonical form LP

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Canonical form LP

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m]$$

Canonical form LP

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m] \qquad x_i \geq 0, i \in [n]$$

Canonical form LP

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m] \qquad x_i \geq 0, i \in [n]$$

What **can** be described by this formulation?

1. Minimization: just maximize $-\sum_i c_i x_i$
2. Equality constraints
    - If you want $\sum_i a_{ij} x_i = b_j$, replace by:
    - $\sum_i a_{ij} x_i \leq b_j$
    - $-\sum_i a_{ij} x_i \leq -b_j$
3. Free variables: unconstrained x can be represented as $x^+ - x^-$ where $x^+ \geq 0$ and $x^- \geq 0$.

Canonical form LP

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m] \qquad x_i \geq 0, i \in [n]$$

What **can** be described by this formulation?

1. Minimization: just maximize $-\sum_i c_i x_i$
2. Equality constraints
   - If you want $\sum_i a_{ij} x_i = b_j$, replace by:
   - $\sum_i a_{ij} x_i \leq b_j$
   - $-\sum_i a_{ij} x_i \leq -b_j$
3. Free variables: unconstrained x can be represented as $x^+ - x^-$ where $x^+ \geq 0$ and $x^- \geq 0$.

What **cannot**?

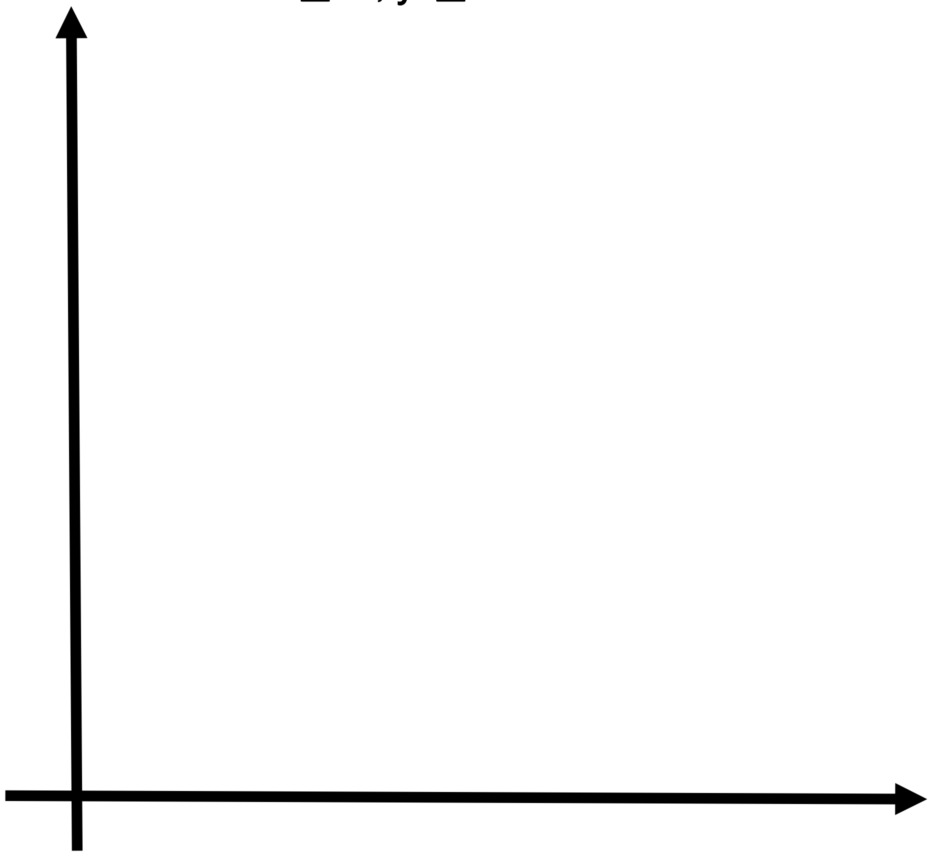
1. Strict inequalities: $x > 0$ is not allowed
2. Arbitrary nonlinear constraints and objectives
3. Integer variables

LP: example and terminology

- Maximize $x + y$
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$

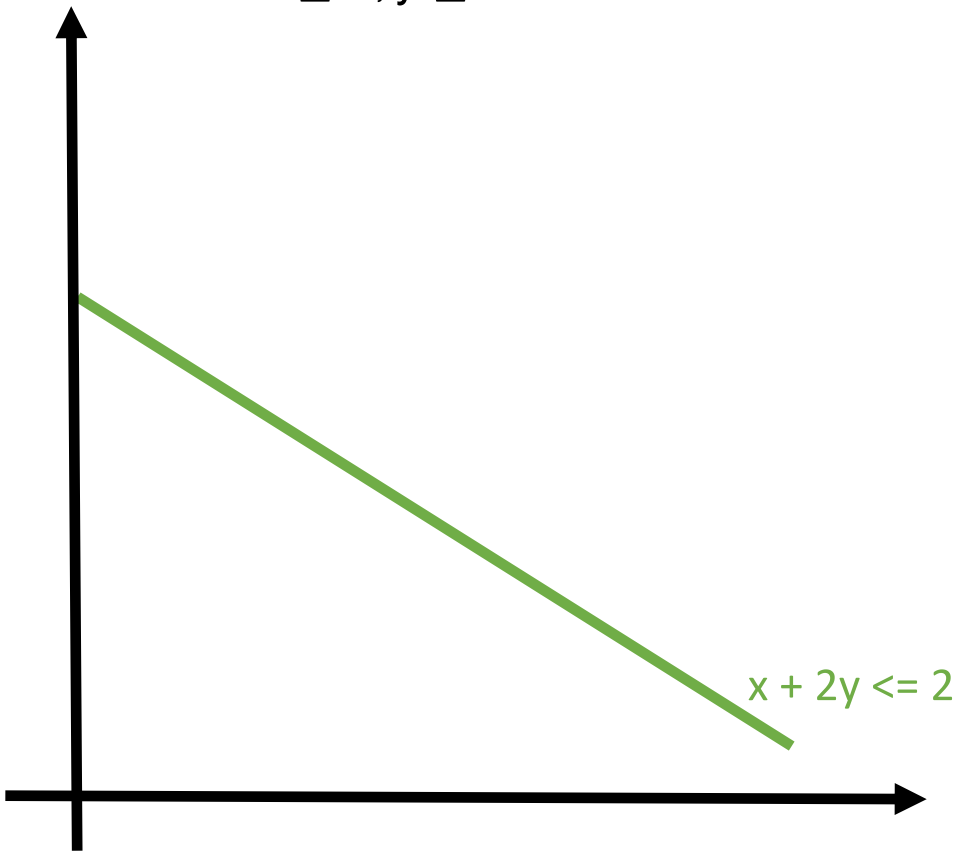
LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$



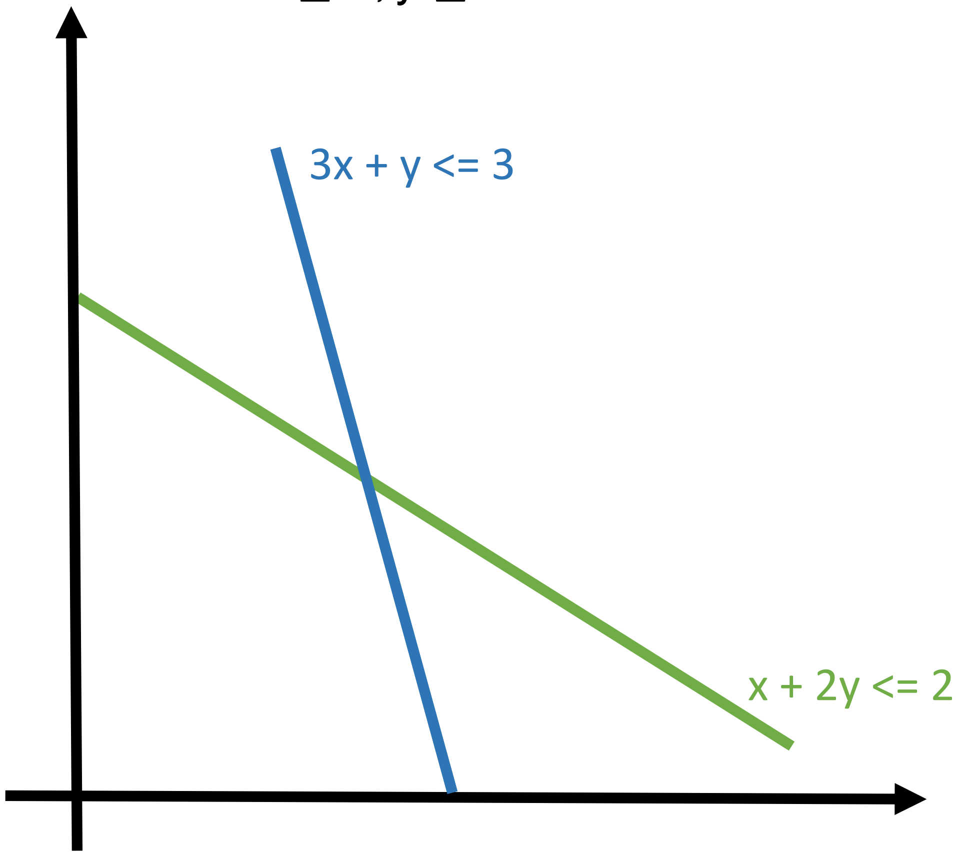
LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$



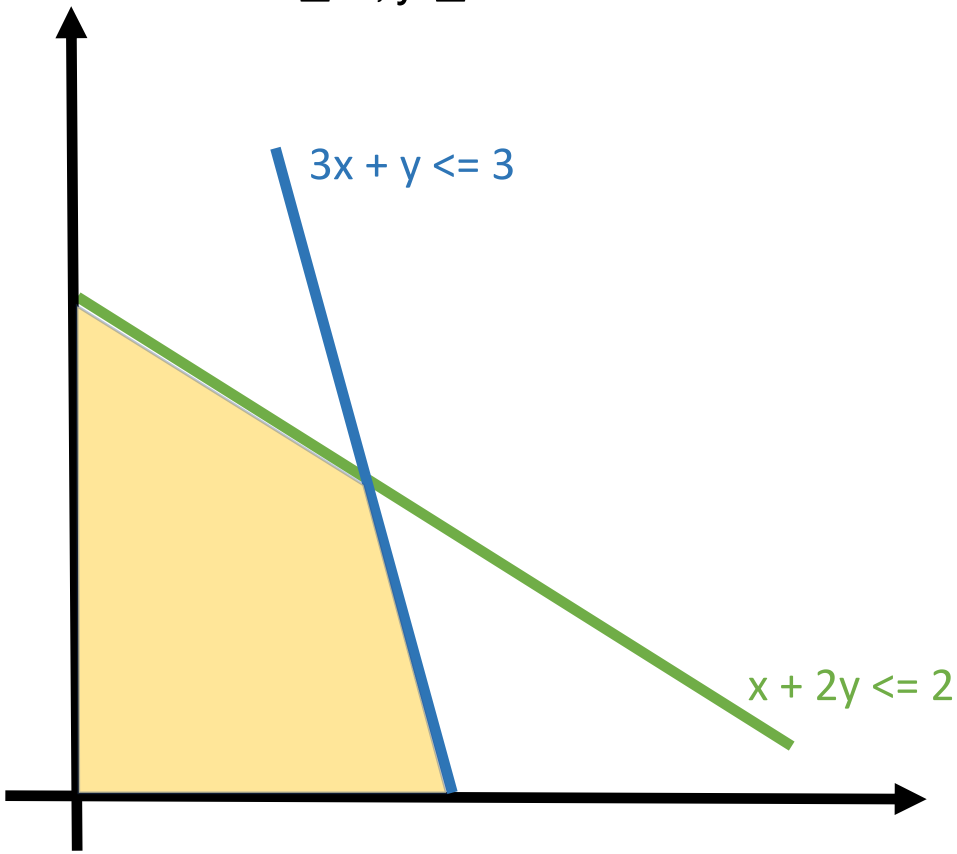
LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$



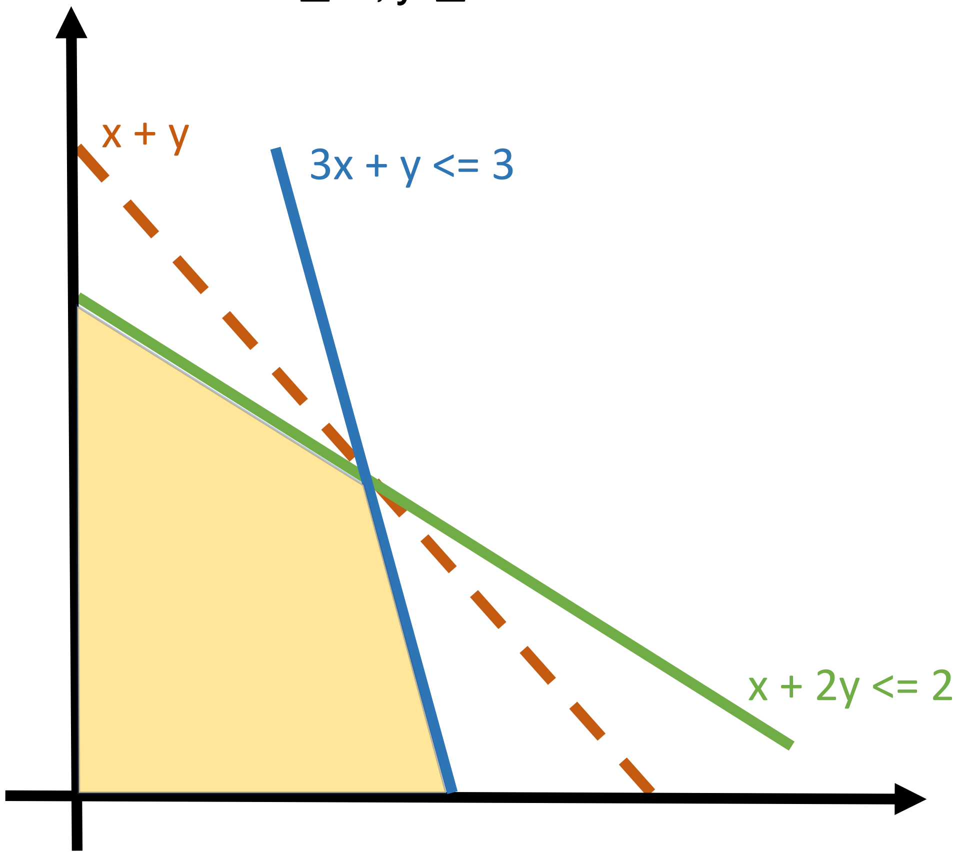
LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$



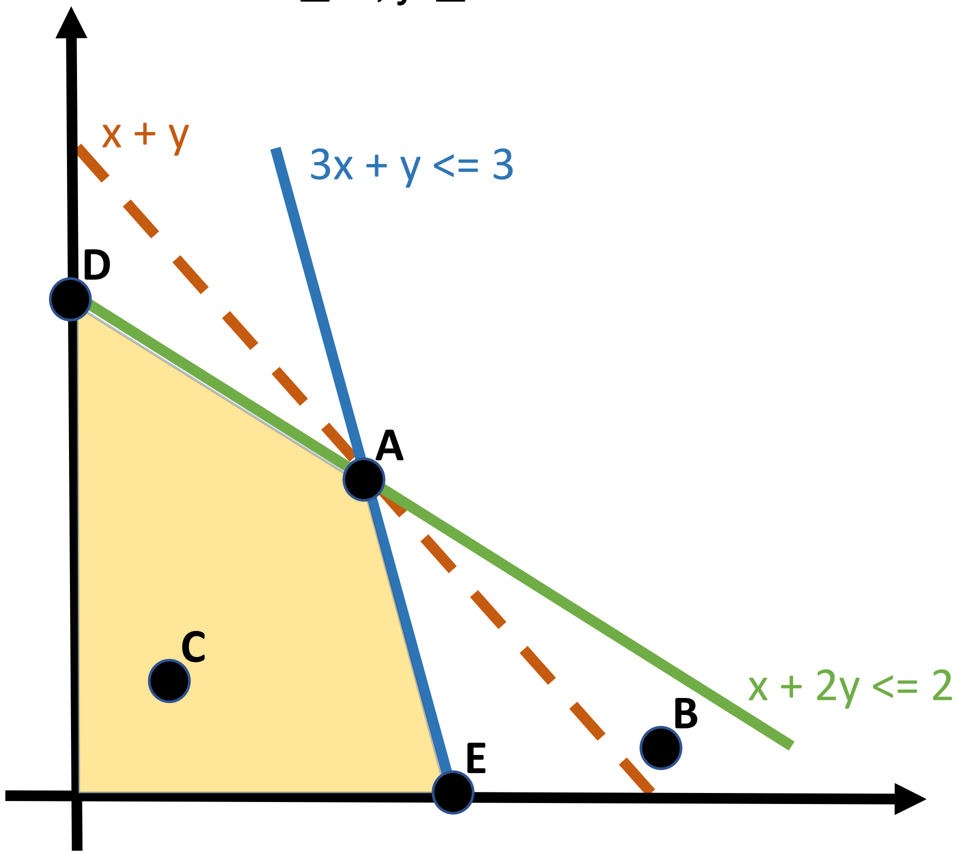
LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$



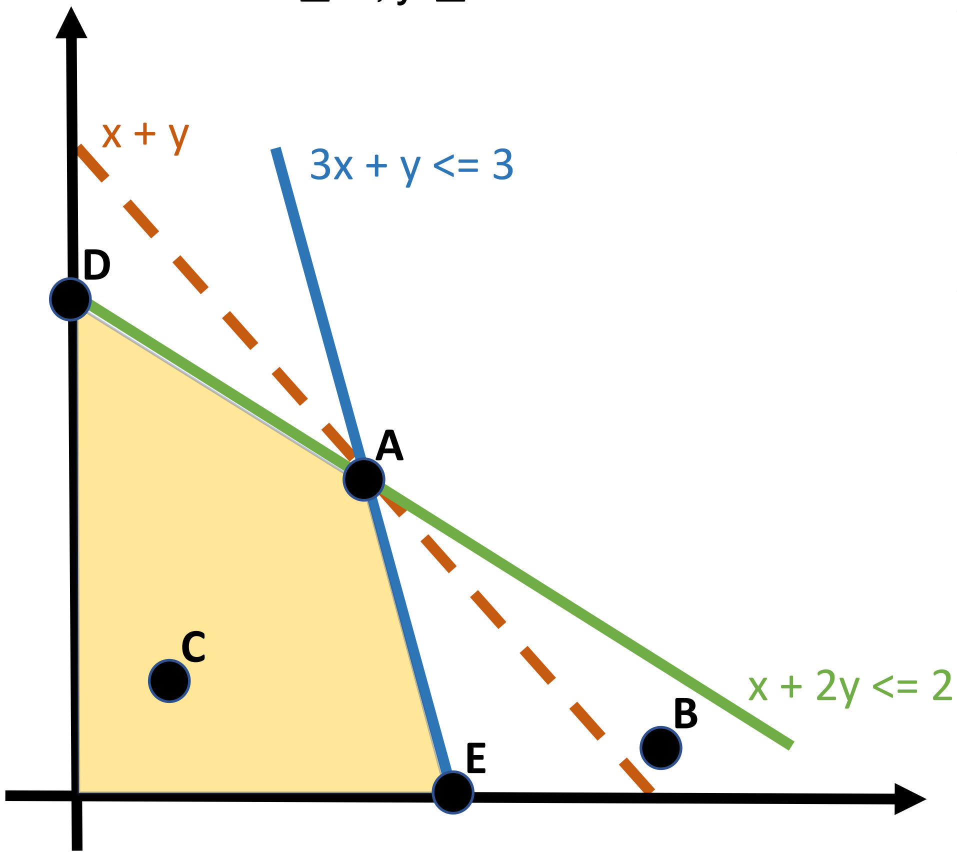
LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$



LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$

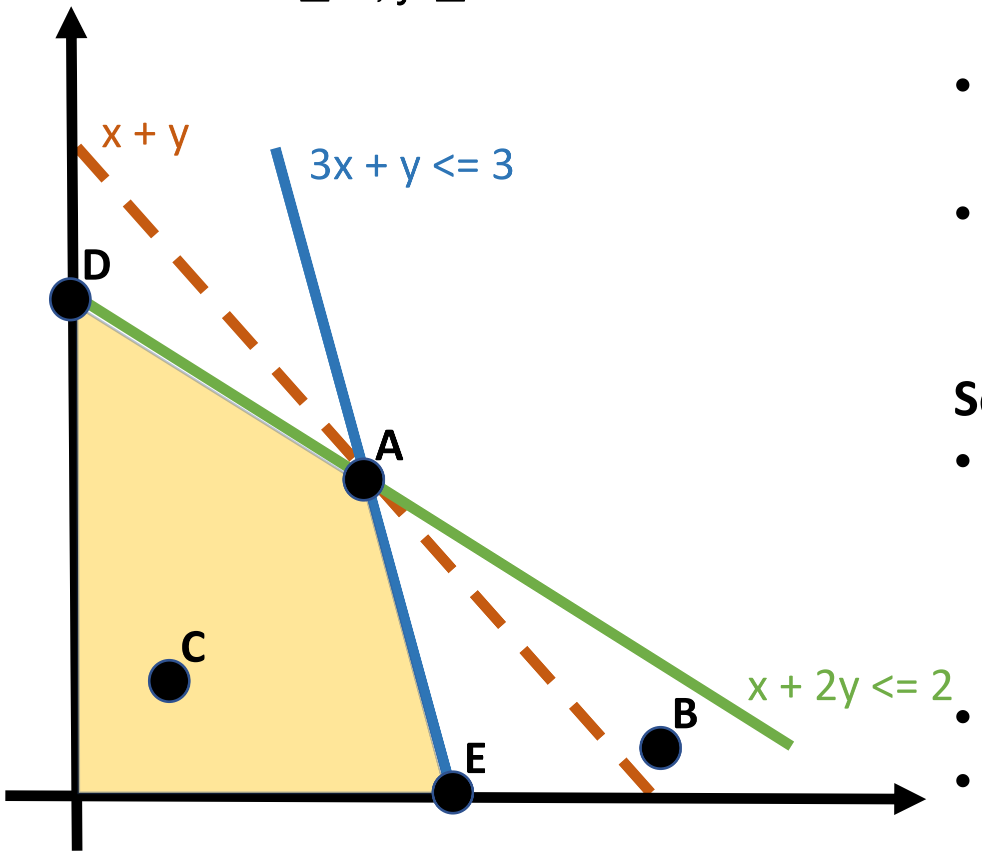


Terminology

- **Feasible point:** a point that satisfies the constraints
  - **Example:** A, C, D, E
- **Infeasible point:** a point that is not feasible
  - **Example:** B
- **Feasible region:**
  - Set of all feasible points (yellow)
- **Extreme point:** the one at the boundary of feasible region
  - **Example:** A, D, E
- **Binding constraint:** the one that holds with equality

LP: example and terminology

- Maximize x + y
- Subject to
  - $x + 2y \leq 2$
  - $3x + y \leq 3$
  - $x \geq 0, y \geq 0$



Terminology

- **Feasible point:** a point that satisfies the constraints
  - **Example:** A, C, D, E
- **Infeasible point:** a point that is not feasible
  - **Example:** B
- **Feasible region:**
  - Set of all feasible points (yellow)
- **Extreme point:** the one at the boundary of feasible region
  - **Example:** A, D, E
- **Binding constraint:** the one that holds with equality

Some facts

- Three types of linear programming problems
  - **Infeasible:** no point satisfy the constraints
  - **Feasible:** there is an optimal solution
  - **Unbounded objective:** the objective can be improved infinitely
- If there is a solution, it is always at the extreme point
- Number of non-zeros is at most the number of constraints

Why linear programming?

- Can be solved very efficiently:
  - problems with hundreds of thousands of variables might take a few secs on a laptop

- Is used as a building block for solving more complicated problems

- Flexible and has a lot of applications:

  - Process optimization (chemical engineering, oil industry, manufacturing)

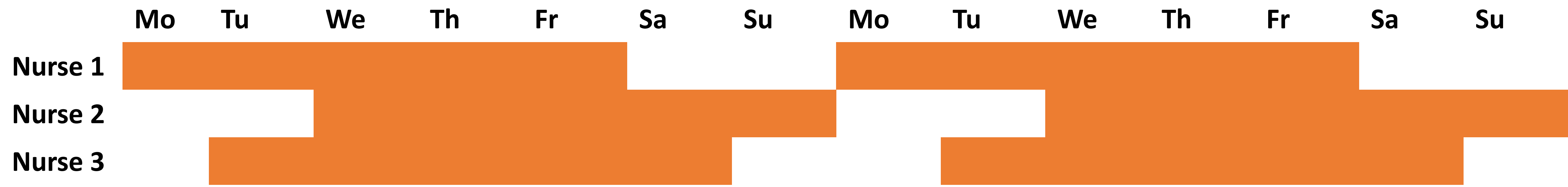
  - Workforce scheduling

  - Revenue management (hotels, airlines, restaurants)

  - Investment

Examples of linear programs: scheduling

Need to schedule nurses. Each nurse goes for a 5-day shift followed by 2 days of rest.



Daily demand for nurses is predictable and is given by the following table

Mo	Tu	We	Th	Fr	Sa	Su
17	13	15	19	14	16	11

Minimize the total number of nurses given the constraints

Examples of linear programs: scheduling

- Decision variables: x_i (how many nurses **start** at day i)
- Objective: minimize $\sum_{i=7}^{n} x_i$
- Subject to constraints
  - $x_1 + x_4 + x_5 + x_6 + x_7 \geq 17$ (Mo)
  - $x_1 + x_2 + x_5 + x_6 + x_7 \geq 13$ (Tu)
  - $x_1 + x_2 + x_3 + x_6 + x_7 \geq 15$ (We)
  - $x_1 + x_2 + x_3 + x_4 + x_7 \geq 19$ (Th)
  - $x_1 + x_2 + x_3 + x_4 + x_5 \geq 14$ (Fr)
  - $x_2 + x_3 + x_4 + x_5 + x_6 \geq 16$ (Sa)
  - $x_3 + x_4 + x_5 + x_6 + x_7 \geq 11$ (Su)
  - $x_i \geq 0, i = 1, \ldots, 7$

Examples of linear programs: investment

- There are n stocks.
- You own s_i shares of stock i.
- You must raise at least C dollars of cash after taxes.
- The expected price of the stock one year from now is r_i
- The capital gains tax is 30%
- The transaction cost for selling stock is 1% of its price
- Example: per-share selling price \$50 and buying price \$30
- Net cash: $(50 - 0.3 \times (50 - 30) - 0.01 \times 50) \times 1,000 = \$43,500$

Formulate a linear programming problem that describes your decision

Examples of linear programs: investment

- Decision variables: x_i (how much of stock i to sell)
- Objective: value of the portfolio at the end of the year
  - Maximize $\sum_{i=1}^{n} r_i(s_i - x_i)$
- Subject to constraints
  - Need to raise C dollars in cash: $\sum_{i=1}(p_i - 0.3(p_i - q_i) - 0.01p_i)x_i \geq C$
  - Can’t sell more than you have: $0 \leq x_i \leq s_i$

Examples of linear programs: manufacturing

- n products, m raw materials
- c_j: profit per unit sold of product j
- b_i: units available of material i
- a_{ij}: units of material i needed to produce one unit of product j

Examples of linear programs: manufacturing

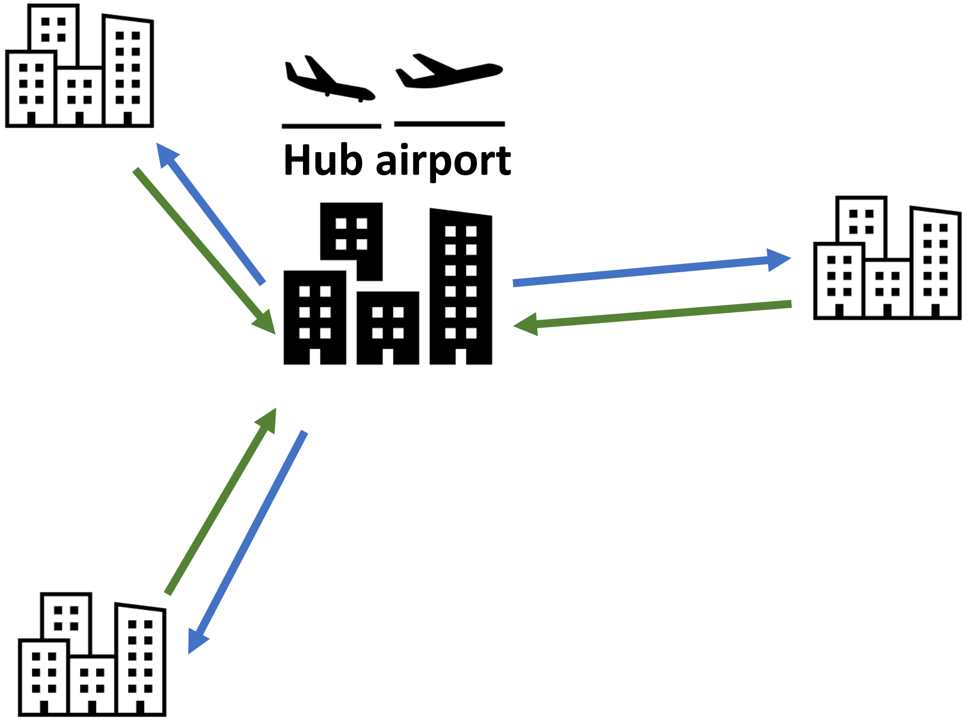
- **Variables:** x_j (amount of product j produced)
- **Objective:** maximize total profit $\sum_j c_j x_j$
- **Constraints:**
  - $\sum_j a_{ij} x_j \leq b_i$ for all i. (Material constraints)
  - $x_j \geq 0$

Examples of linear programs: manufacturing

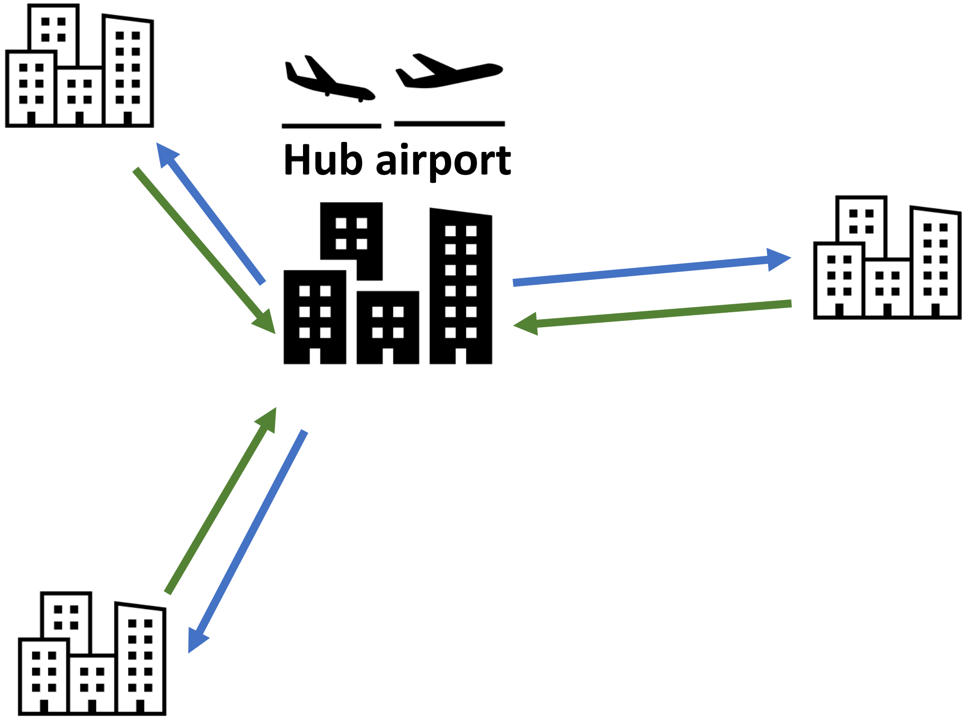
- **Variables:** x_j (amount of product j produced)
- **Objective:** maximize total profit $\sum_j c_j x_j$
- **Constraints:**
  - $\sum_j a_{ij} x_j \leq b_i$ for all i. (Material constraints)
  - $x_j \geq 0$

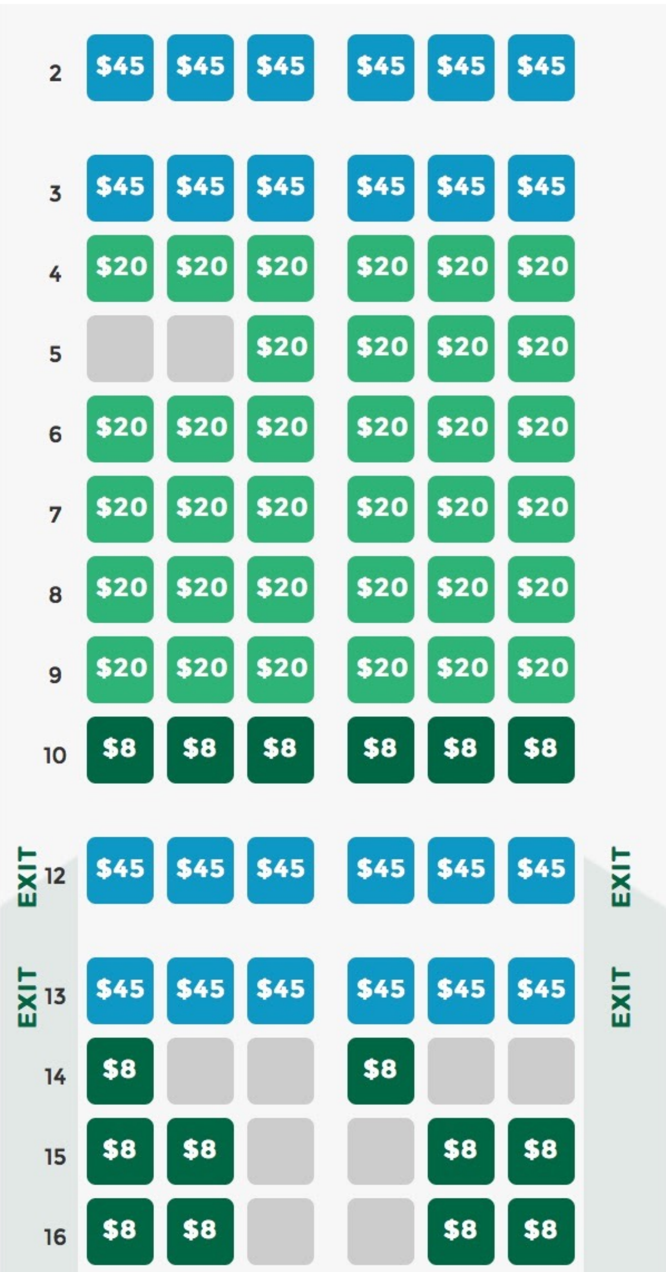
More complicated extensions of this model serve as basis for Material Resource Planning and Enterprise Resource Planning systems

Examples of linear programs: revenue management

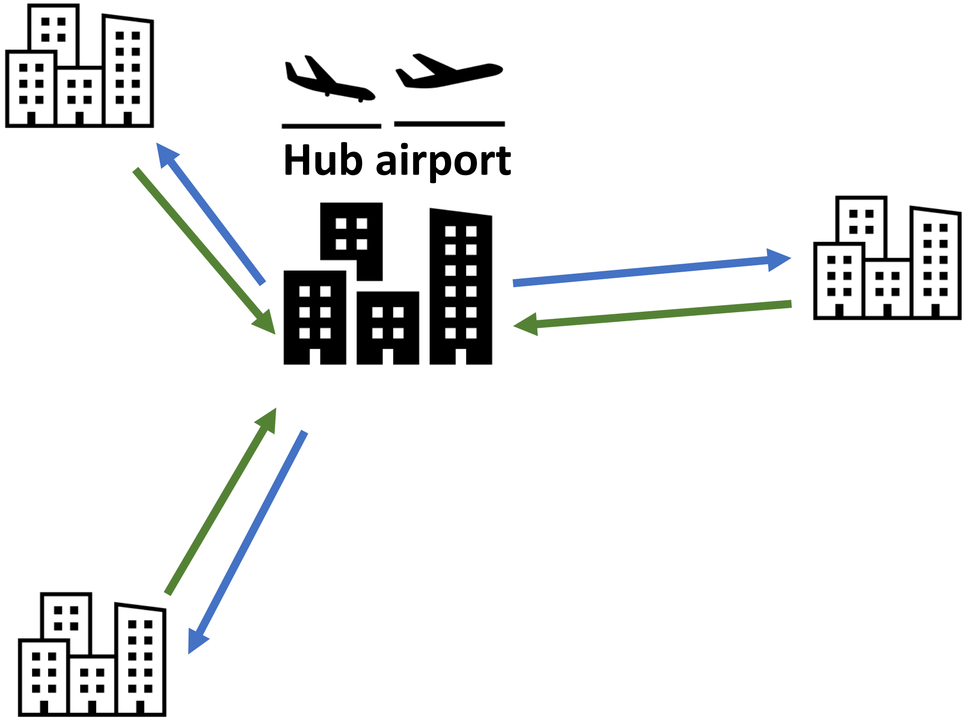


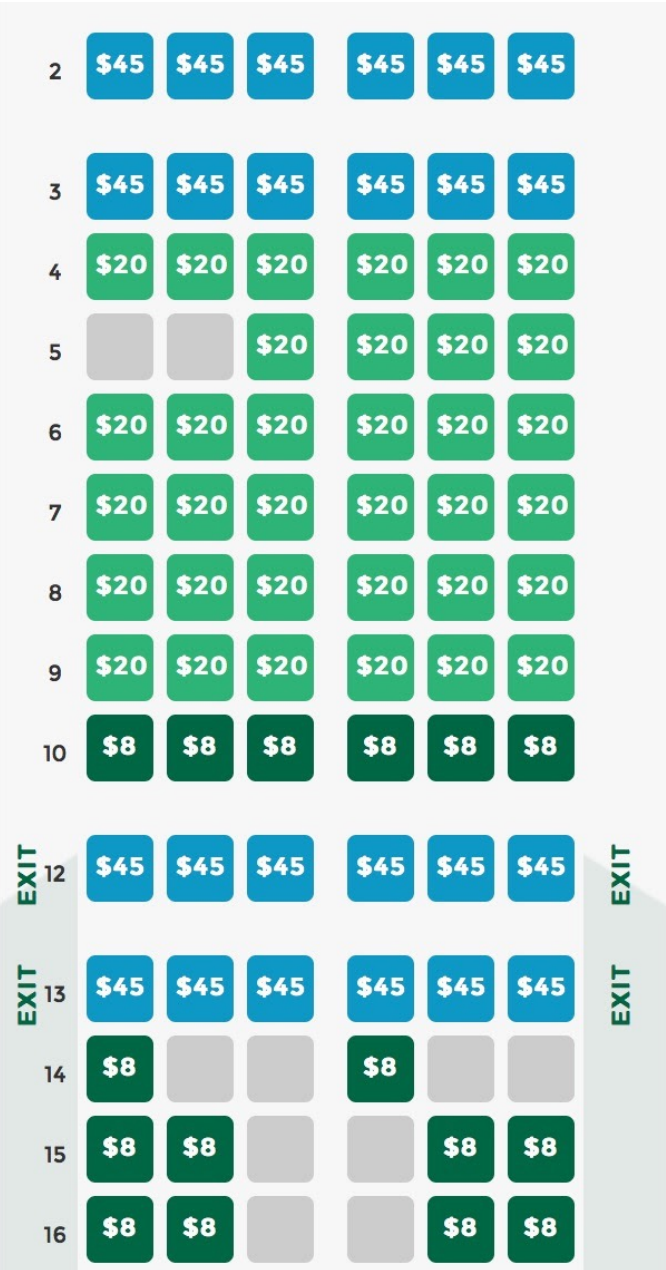
Examples of linear programs: revenue management





Examples of linear programs: revenue management



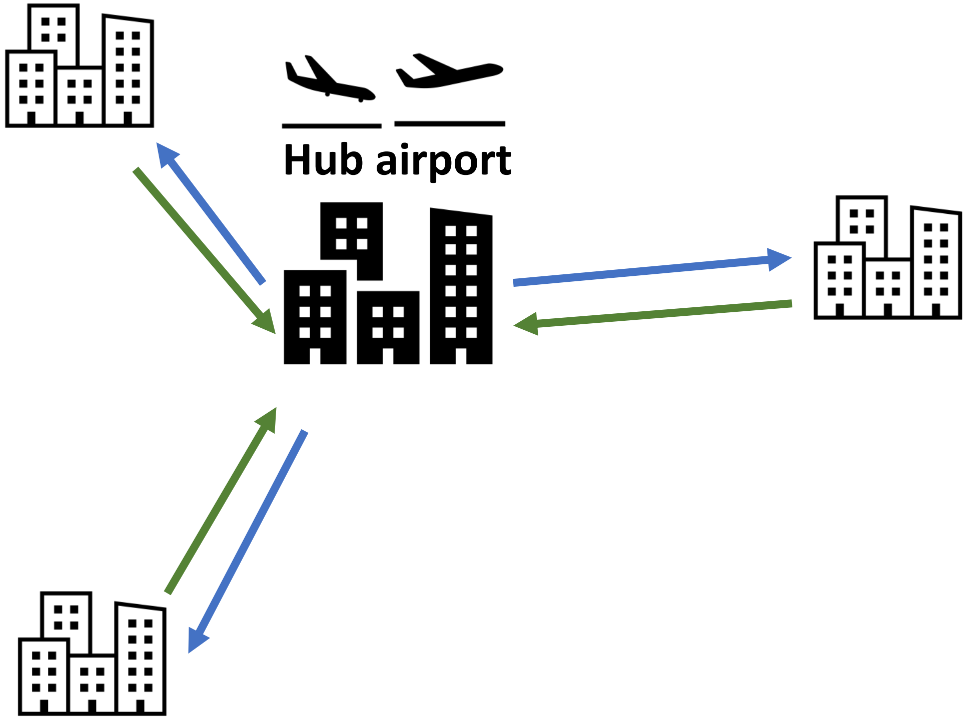


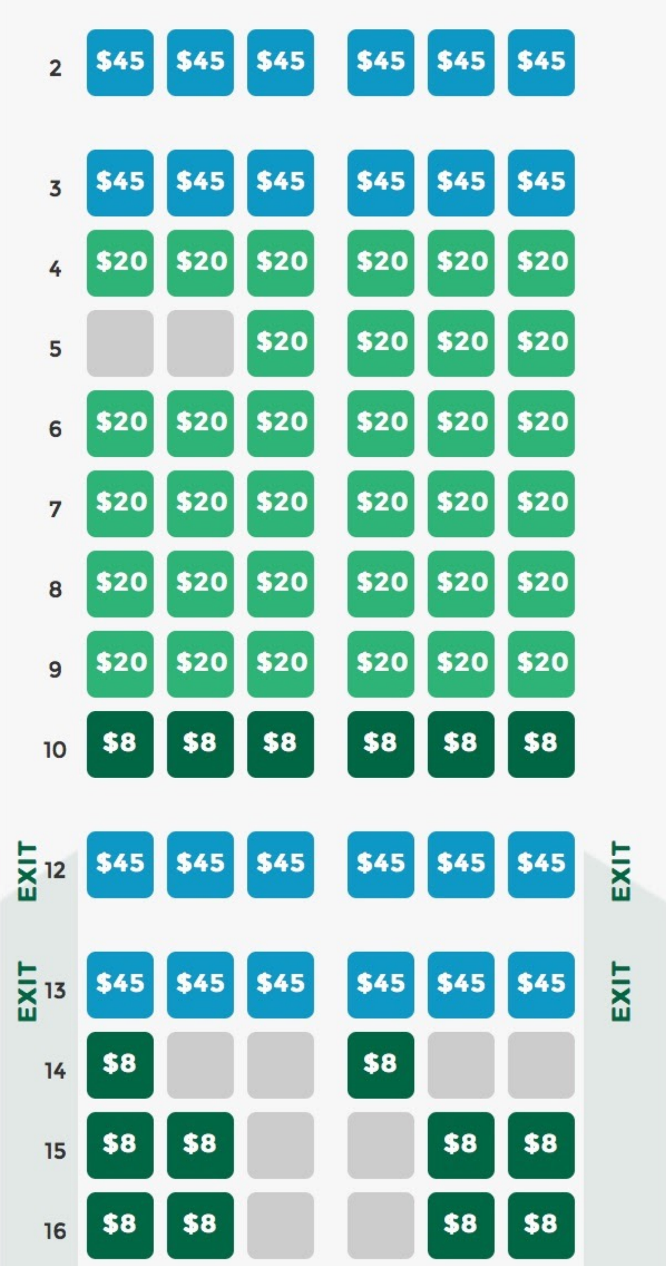
Economy seats—not the same price

Some are refundable
Some require return tickets

Example: Q and Y fares
Qs are more expensive than Y

Examples of linear programs: revenue management





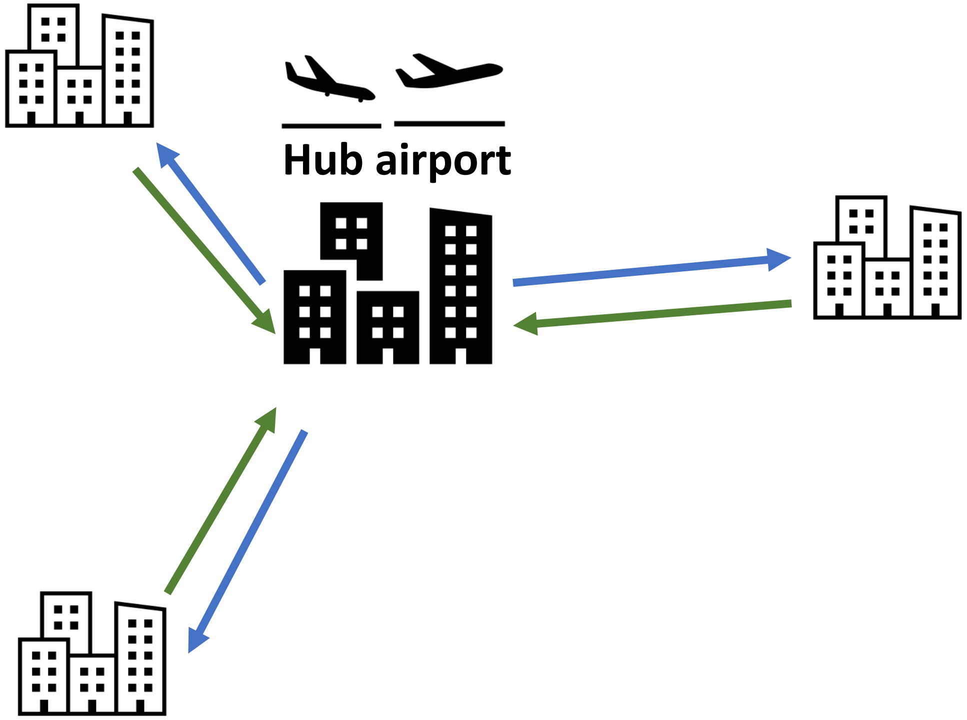
How to split between two fare classes?

Economy seats—not the same price

Some are refundable
Some require return tickets

Example: Q and Y fares
Qs are more expensive than Y

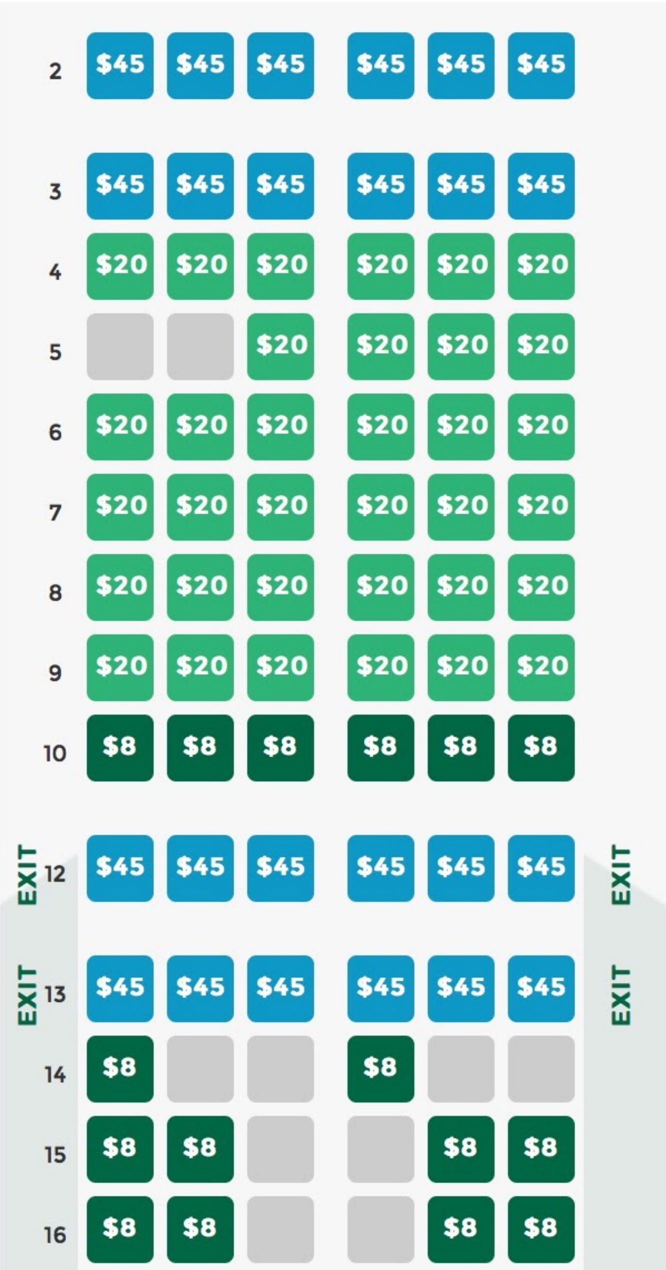
Examples of linear programs: revenue management



Economy seats—not the same price

Some are refundable
Some require return tickets

Example: Q and Y fares
Qs are more expensive than Y



How to split between two fare classes?

Very important question!

Am. Air. CEO circa 1985:

“We estimate that revenue management has generated **$1.4 billion in incremental revenue** in the last three years. “

Examples of linear programs: revenue management

- n origins, n destinations
- 1 airport hub
- 2 fare classes: Q and Y
- Revenues for each ride and class: r_{ij}^{Q} and r_{ij}^{Y}
- Leg capacities
  - C_{i0} for inbound flights
  - C_{0j} for outbound flights
- Expected demands: D_{ij}^{Q}, D_{ij}^{Y}

Examples of linear programs: revenue management

- n origins, n destinations
- 1 airport hub
- 2 fare classes: Q and Y
- Revenues for each ride and class: r_{ij}^{Q} and r_{ij}^{Y}
- Leg capacities
  - C_{i0} for inbound flights
  - C_{0j} for outbound flights
- Expected demands: D_{ij}^{Q}, D_{ij}^{Y}

- **Variables:** Q_{ij} and Y_{ij} (tickets sold for $i \to j$ flights)
- **Objective:** maximize total profit $\sum r_{ij} Q_{ij} + r_{ij}^{Y} Y_{ij}$
- **Constraints:**
  - $\sum_j a_{ij}(Q_{ij} + Y_{ij}) \leq C_{i0}$ for all i. (Inbound flights)
  - $\sum_i a_{ij}(Q_{ij} + Y_{ij}) \leq C_{0j}$ for all i. (Outbound flights)
  - $0 \leq Q_{ij} \leq D_{ij}^{Q}, 0 \leq Y_{ij} \leq D_{ij}^{Y}$

What else can we represent with LP? Minimax

Objective

$$\min_{x_j} \left(\max_i \sum_j c_{ij} x_j\right)$$

Subject to some linear constraints

Can we make it a linear programming problem? Yes!

$$\min_{x_j, z} z$$

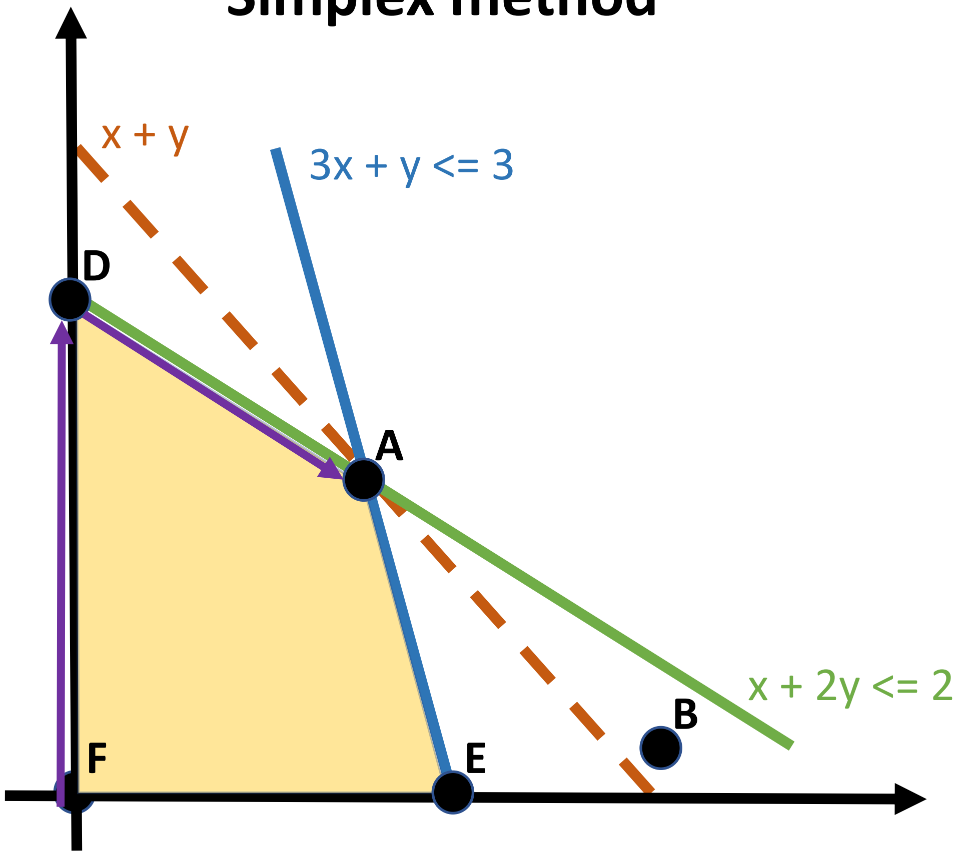
subject to

$$\sum_j c_{ij} x_j \leq z \text{ for all } i.$$

Why do we want it? Worst-case profits, piecewise-linear functions, etc.

How do optimization algos work

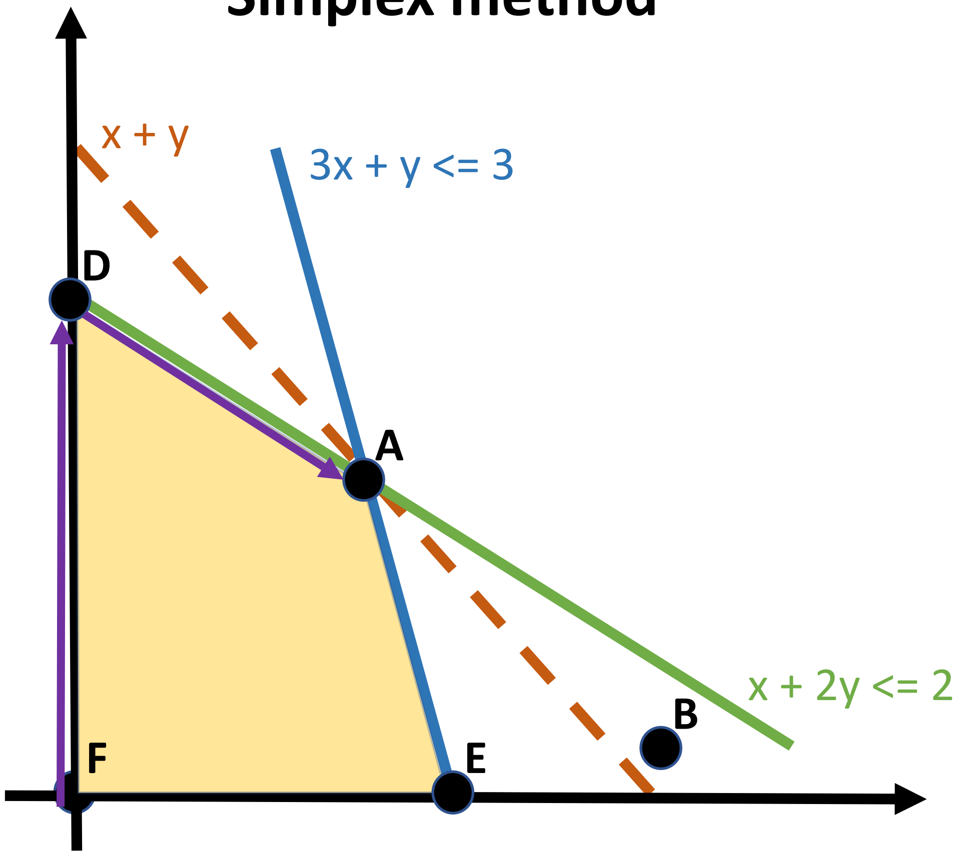
Simplex method



Move from one extreme point to another
Worst-case can be slow, but typically is fast

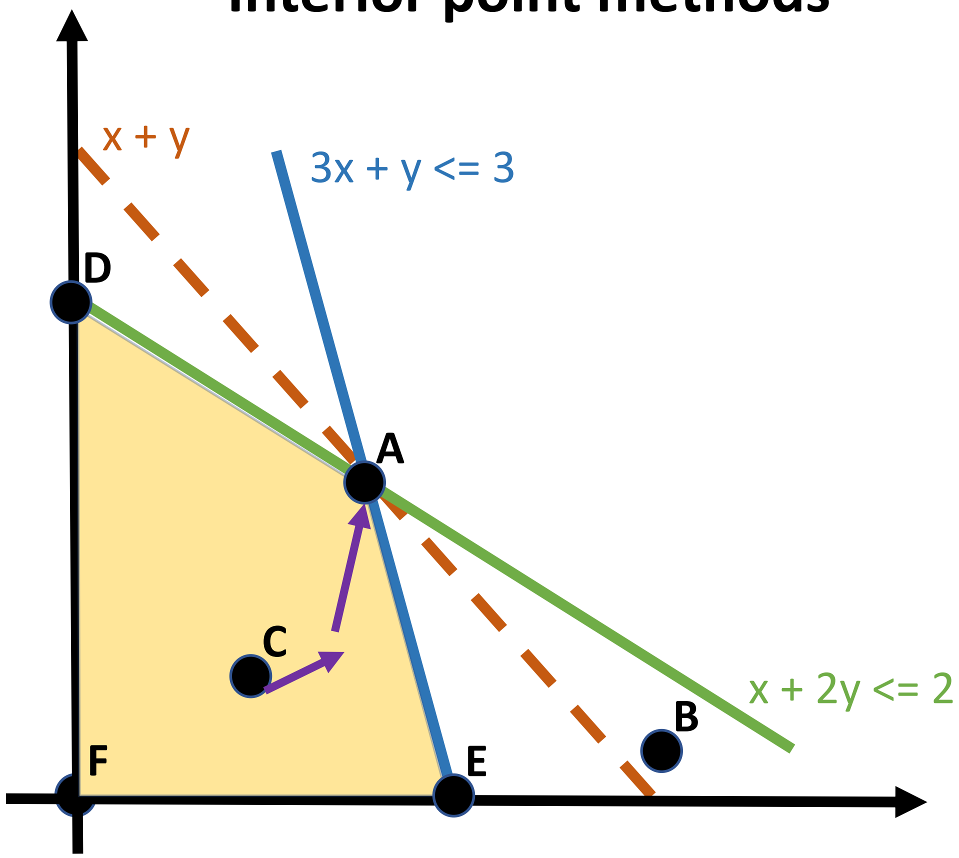
How do optimization algos work

Simplex method



Move from one extreme point to another
Worst-case can be slow, but typically is fast

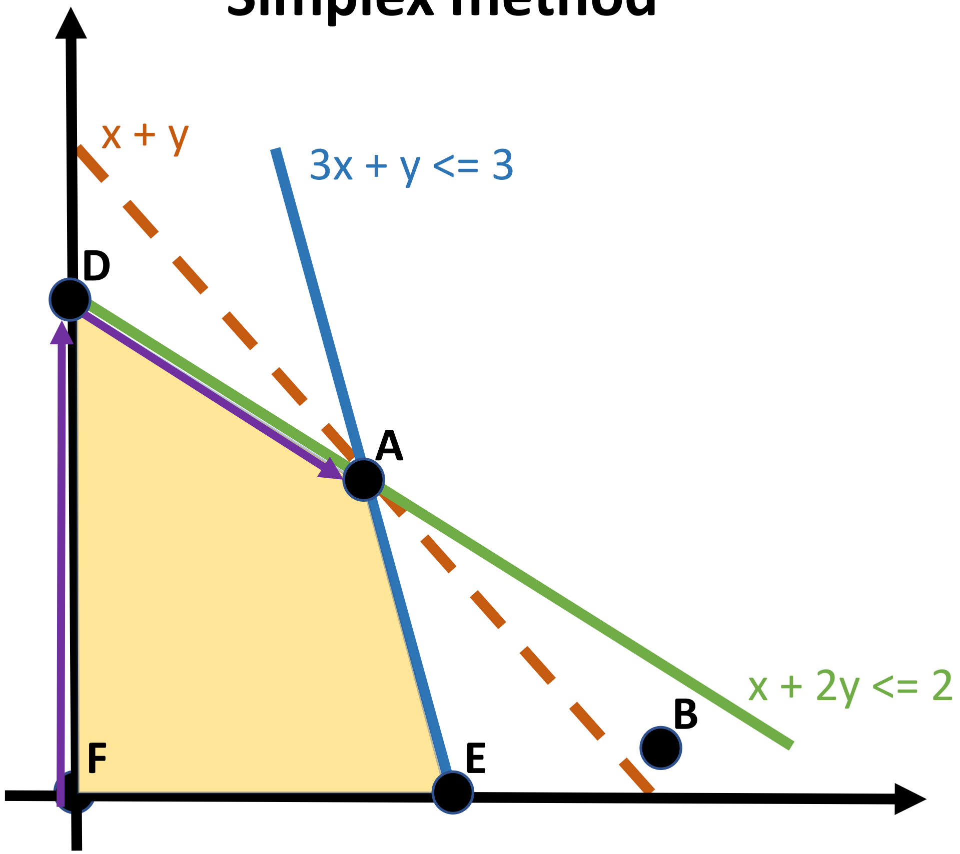
Interior point methods



Move inside the feasible region
Worst-case is provably faster than simplex

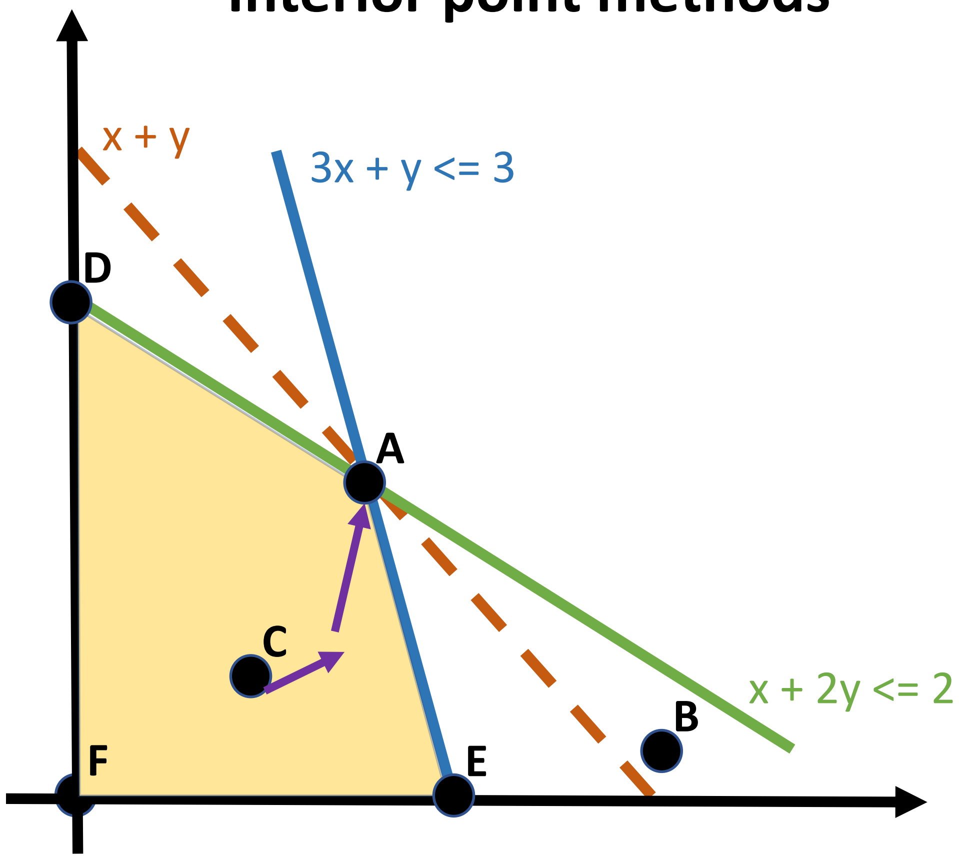
How do optimization algos work

Simplex method



Move from one extreme point to another
Worst-case can be slow, but typically is fast

Interior point methods



Move inside the feasible region
Worst-case is provably faster than simplex

We will not go in detail on the algorithms.
Solvers can create a solution just from problem description.
Commercial solvers: CPLEX, Gurobi. Open-source ones: CBC, GLPK

How does the code look like?

```
import cvxpy as cp
n = 3
C = cp.Variable(nonneg = True)
S = cp.Variable(nonneg = True)
I = cp.Variable(nonneg = True)
objective = cp.Maximize(C + S + I)
constraints = [3*(C+S+I) <= 60,
20*C <= 120,
15*(C+S) <= 180,
8*I <= 120,
2*(C+S+I) <= 60,
C <= 3, S <= 11, I <= 4,
C/3 == S/11,
S/11 == I/4
]
prob = cp.Problem(objective, constraints)
result = prob.solve()
C.value, S.value, I.value (2.571428571474969, 9.428571428741536, 3.428571428633301)
objective.value 15.428571428849807
```

How does the code look like?

```python
import cvxpy as cp
n = 3
C = cp.Variable(nonneg = True)
S = cp.Variable(nonneg = True)
I = cp.Variable(nonneg = True)
objective = cp.Maximize(C + S + I)
constraints = [3*(C+S+I) <= 60,
20*C <= 120,
15*(C+S) <= 180,
8*I <= 120,
2*(C+S+I) <= 60,
C <= 3, S <= 11, I <= 4,
C/3 == S/11,
S/11 == I/4
]
prob = cp.Problem(objective, constraints)
result = prob.solve()
C.value, S.value, I.value
```

(2.571428571474969, 9.428571428741536, 3.428571428633301)

```python
objective.value
```

15.428571428849807

Variables

Objective

Constraints

Recap

- We’ve learned how to specify linear programs
  - Variables
  - Linear objective
  - Linear constraints

- We’ve seen how it’s applied to business problems

- We’ve also seen some tricks on using it

- You can now start formulating LPs for problems 1 and 2 in HA 1

Up next

- How do we solve linear programming problems in code?

- What happens when some variables need to be integer?
  - E. g. what happens if we can only produce discrete units

- How do we incorporate various logical constraints?
  - E. g. suppose that we want to have an OR of two constraints?

Business analytics I
Operations Analytics
Class 3
Integer Programming Formulations

Marat Salikhov
March 21^{st}, 2022

Recap: linear programming

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m] \qquad x_i \geq 0, i \in [n]$$

Recap: linear programming

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m] \qquad x_i \geq 0, i \in [n]$$

x_i integer or boolean (either 0 or 1)

Recap: linear programming

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m] \qquad x_i \geq 0, i \in [n]$$

x_i integer or boolean (either 0 or 1)

Why?

- Domain requirements
  - Discrete price points: $1, $5, $10
  - No. of units produced: 1, 2, 3, …
  - Do you invest in a project or not? (0 or 1)

- Encode logical constraints
  - "You can't run both projects at the same time"

Recap: linear programming

$$\max_{x_i} \sum_{i=1}^{n} c_i x_i$$

Subject to constraints

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j, j \in [m] \qquad x_i \geq 0, i \in [n]$$

x_i integer or boolean (either 0 or 1)

Why?

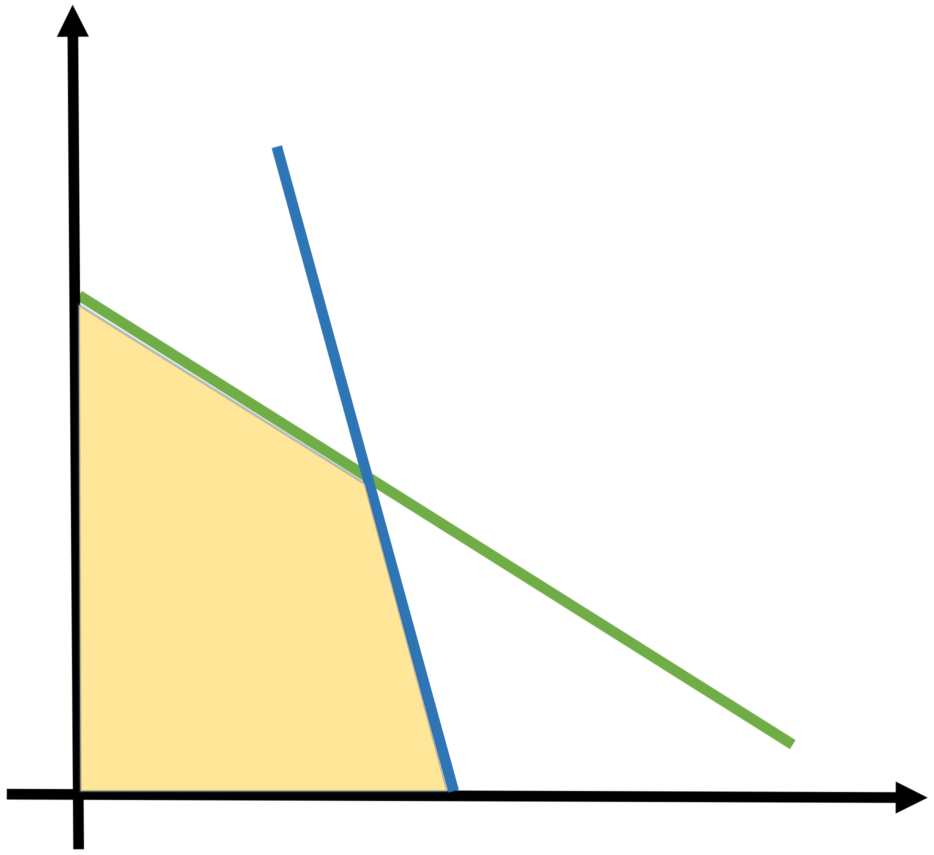
- Domain requirements
  - Discrete price points: \$1, \$5, \$10
  - No. of units produced: 1, 2, 3, …
  - Do you invest in a project or not? (0 or 1)
- Encode logical constraints
  - "You can't run both projects at the same time"

Types of integer programs

- "Pure" integer programs
  - Only integer variables allowed
- Mixed integer programs (MIP)
  - Both integer and continuous vars

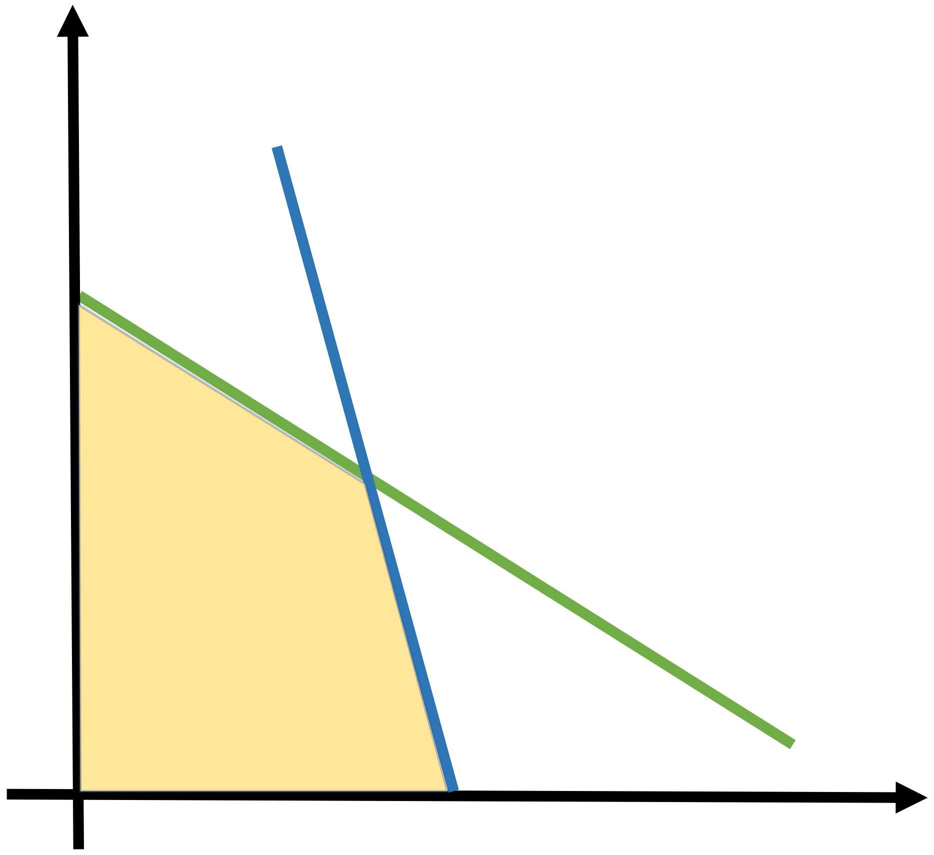
Integer vs. linear programming: graphics

Linear programming

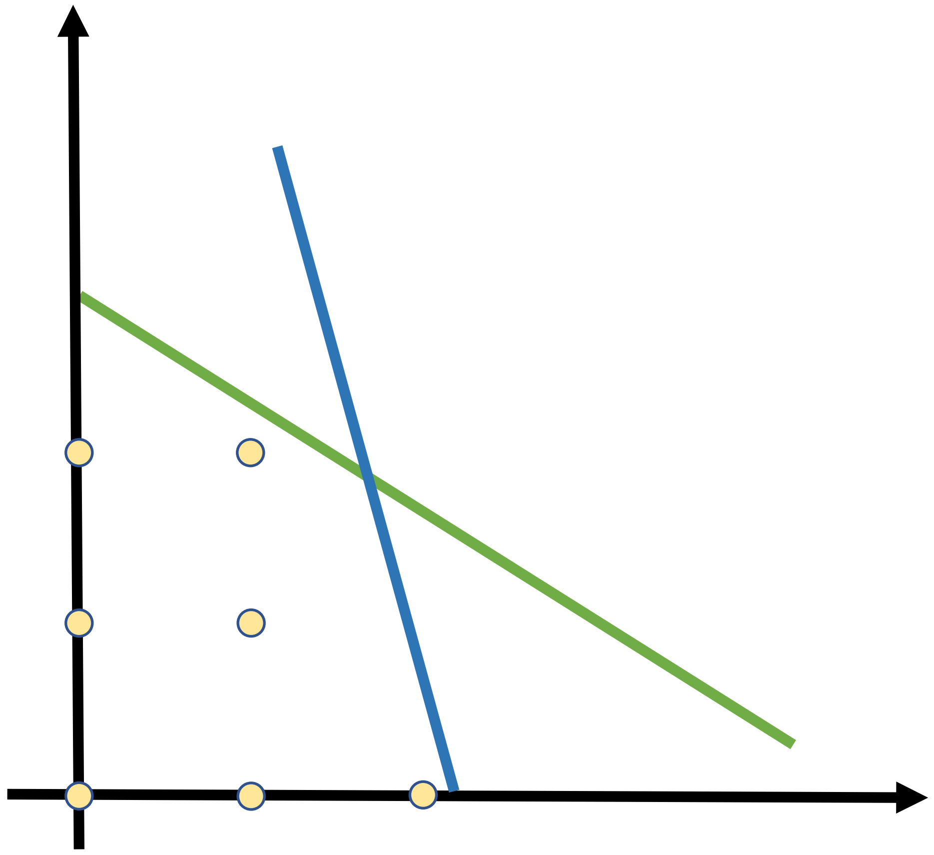


Integer vs. linear programming: graphics

Linear programming

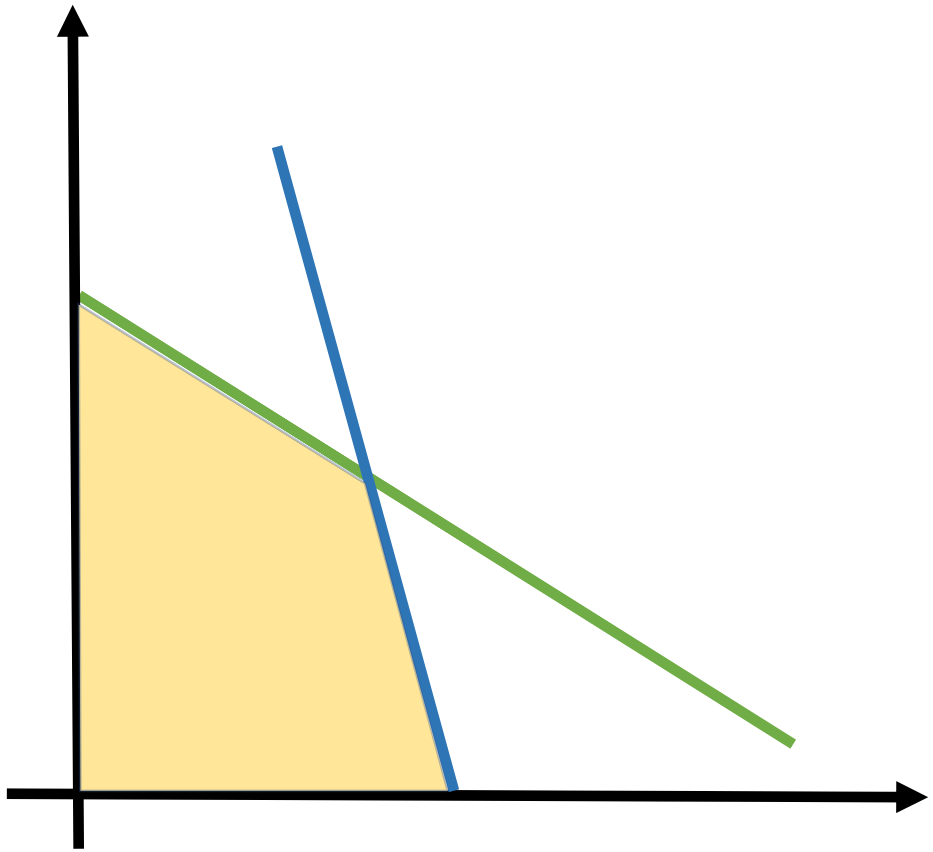


Integer programming

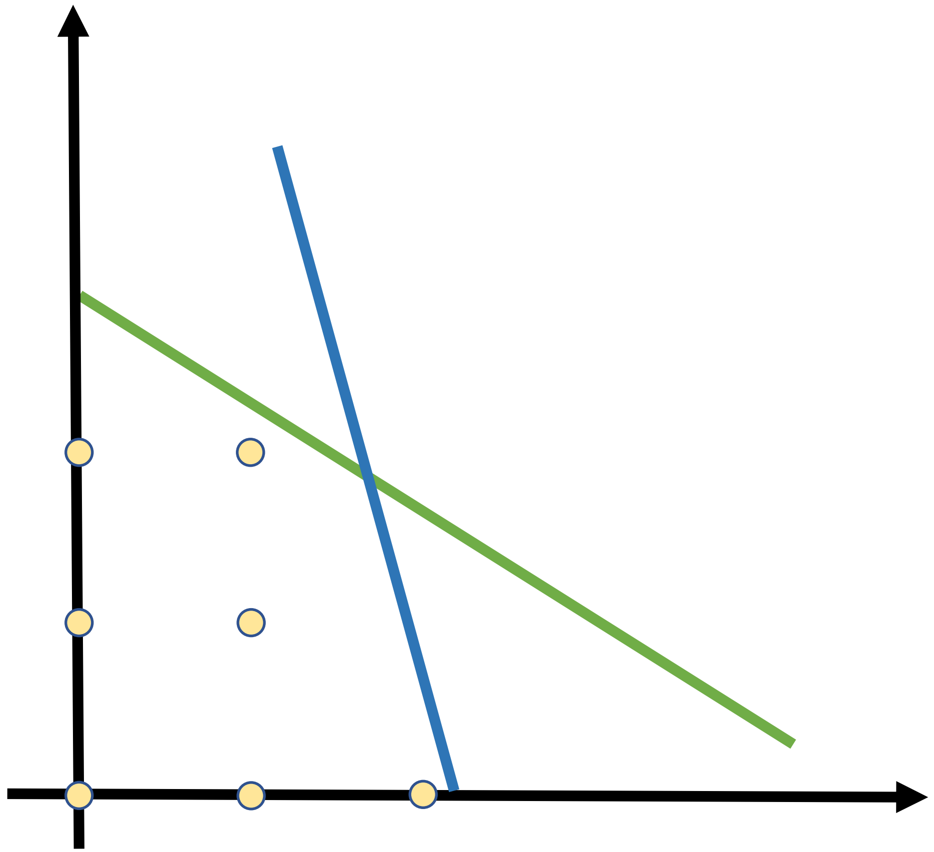


Integer vs. linear programming: graphics

Linear programming



Integer programming



IP might be much more difficult to solve than LP.

Clever algorithms exist. (We would need a whole course to cover them.)

Tractability of integer programming

- Some integer programming problems are **extremely simple**

$$\max_{x_i} \sum_i a_i x_i \text{ subject to } \sum_i x_i = 1,\ x_i \in \{0,1\}\ \forall i$$

- **Others are extremely hard**
  - In general, integer programming problems are NP-complete
  - Roughly speaking, that means that its unknown whether an efficient algo exists
  - Example of such problem: traveling salesman problem (TSP)
    - Take a shortest tour of N cities while visiting each city only once
  - Another example: Boolean satisfiability (SAT)
    - Is a Boolean formula consisting of variables, ANDs, ORs, and NOTs ever true?

- Many IP problems that arise in practice are quite tractable
  - There's also tremendous progress in comp. power and algos:
    - $10^{10}\times$ speedup from 1990 to 2010. 400 years then, 1 second now.
    - For example, this speedup has completely changed sports scheduling
  - **So now it's a great time to see how integer programming can be applied.**

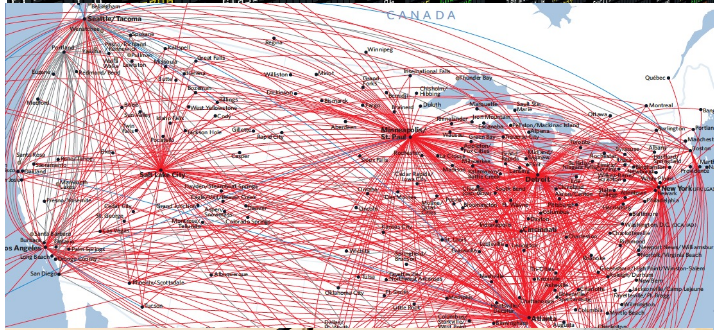
Applications of integer programming

- **Assignment and scheduling**
  - Assigning workers to shifts or machines to jobs
  - A worker is either assigned to a shift or not

- **Vehicle routing**
  - How should delivery trucks be scheduled?

- **Budgeting**
  - How to allocate a finite budget over a set of projects?
  - A project is either initiated or not

- **Facility location**
  - Which locations to choose for a factory, warehouse, etc.



Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design				
DB design				
API design				
Testing plan				

Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h			
DB design	10h			
API design	12h			
Testing plan	5h			

Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h		
DB design	10h	4h		
API design	12h	8h		
Testing plan	5h	6h		

Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h	8h	
DB design	10h	4h	5h	
API design	12h	8h	3h	
Testing plan	5h	6h	5h	

Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h	8h	4h
DB design	10h	4h	5h	11h
API design	12h	8h	3h	12h
Testing plan	5h	6h	5h	7h

Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h	8h	4h
DB design	10h	4h	5h	11h
API design	12h	8h	3h	12h
Testing plan	5h	6h	5h	7h

Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h	8h	4h
DB design	10h	4h	5h	11h
API design	12h	8h	3h	12h
Testing plan	5h	6h	5h	7h

Goal: minimize total time spent

Formulation

$$\min \sum_{i,j} v_{ij} x_{ij}$$

subject to

$$\sum_{i} x_{ij} = 1$$

$$\sum_{j} x_{ij} = 1$$

$$x_{ij} \in \{0, 1\}.$$

Assignment problem

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h	8h	4h
DB design	10h	4h	5h	11h
API design	12h	8h	3h	12h
Testing plan	5h	6h	5h	7h

Goal: minimize total time spent

Formulation

$$\min \sum_{i,j} v_{ij} x_{ij}$$ Total time spent

subject to

$$\sum_{i} x_{ij} = 1$$ One person per task

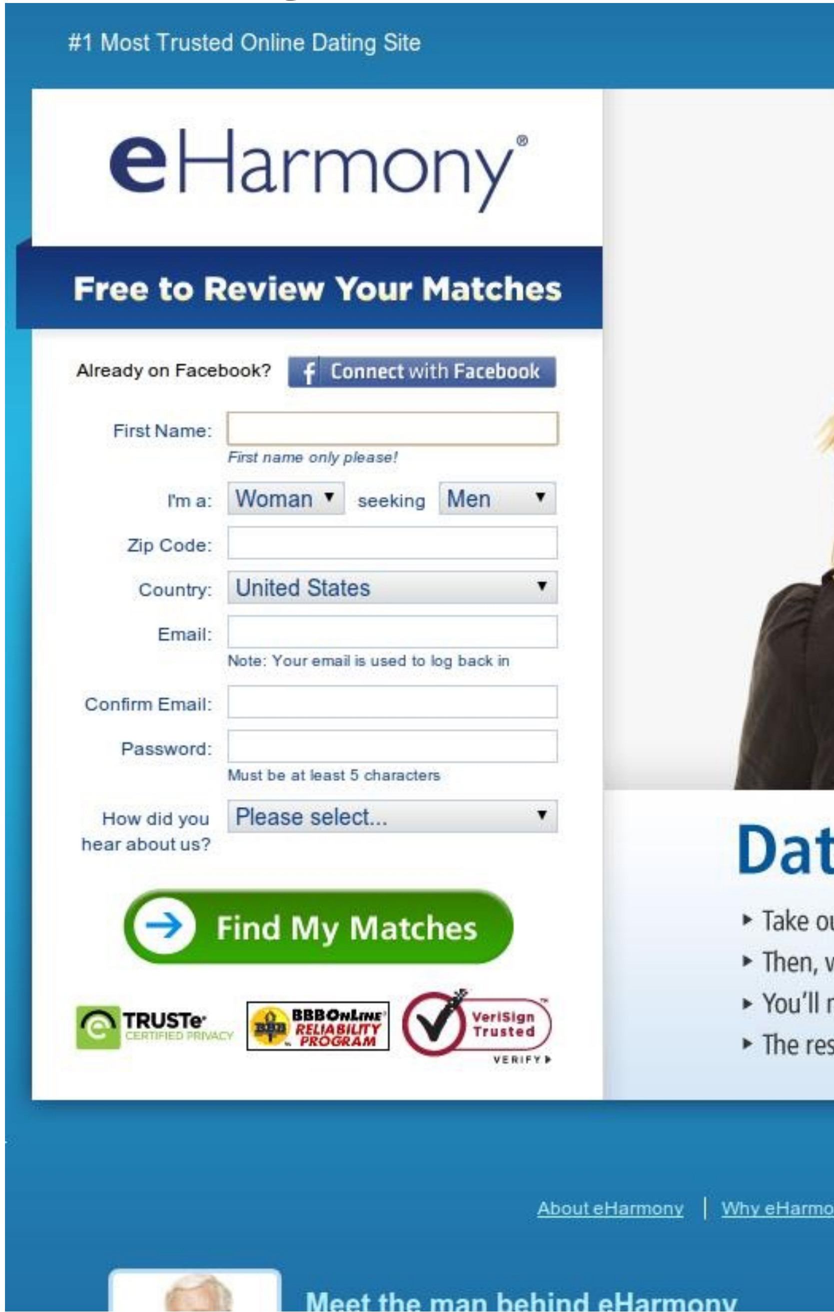
$$\sum_{j} x_{ij} = 1$$ One task per person

$$x_{ij} \in \{0, 1\}.$$ Tasks are given or not

Assignment problem: eHarmony

eharmony

- A dating site
- Works by computing “compatibility score” for each pair
- Then, matches people by solving an integer programming problem
- This problem has several million variables and constraints!
- eHarmony accounts for 4% of all marriages in the US (2012 data)
- Is still quite successful, #2 in the US dating market



Your turn: maximin formulation

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h	8h	4h
DB design	10h	4h	5h	11h
API design	12h	8h	3h	12h
Testing plan	5h	6h	5h	7h

Goal: minimize the slowest time

$$\min_{x_{ij}} \left(\max_{i,j} v_{ij} x_{ij} \right)$$

Your turn: maximin formulation

	Quentin QA guy	Daria DBA	Pilar Programmer	Uter UI guy
UI design	12h	10h	8h	4h
DB design	10h	4h	5h	11h
API design	12h	8h	3h	12h
Testing plan	5h	6h	5h	7h

Goal: minimize the slowest time

$$\min_{x_{ij}} \left(\max_{i,j} v_{ij} x_{ij} \right)$$

$$\min_{x_{ij}} t$$

subject to

$$v_{ij} x_{ij} \leq t \;\; \forall i, j$$

other constraints

Capital budgeting (knapsack) problem

- n projects
- For each project i:
  - Make ”go” or ”no-go” decision x_i
  - The expected benefit is c_i
  - The upfront cash cost is b_i
- Total budget allocated is B

Capital budgeting (knapsack) problem

- n projects
- For each project i:
  - Make "go" or "no-go" decision x_i
  - The expected benefit is c_i
  - The upfront cash cost is b_i
- Total budget allocated is B

$$\max_{x_i} \sum c_i x_i$$

subject to

$$\sum b_i x_i \leq B$$

$$x_i \in \{0, 1\} \ \forall i$$

Capital budgeting (knapsack) problem

- n projects
- For each project i:
  - Make "go" or "no-go" decision x_i
  - The expected benefit is c_i
  - The upfront cash cost is b_i
- Total budget allocated is B

$$\max_{x_i} \sum c_i x_i$$

subject to

$$\sum b_i x_i \leq B$$

$$x_i \in \{0, 1\} \; \forall i$$

Capital budgeting (knapsack) problem

- n projects
- For each project i:
  - Make "go" or "no-go" decision x_i
  - The expected benefit is c_i
  - The upfront cash cost is b_i
- Total budget allocated is B

$$\max_{x_i} \sum c_i x_i$$
subject to
$$\sum b_i x_i \leq B$$
$$x_i \in \{0, 1\} \; \forall i$$

This is the simplest possible capital budgeting problem.

Often, we have constraints saying that some projects are not compatible with each other, or that some of them have to be done as a bundle.

In general, we need logical constraints: and we can implement them in IP.

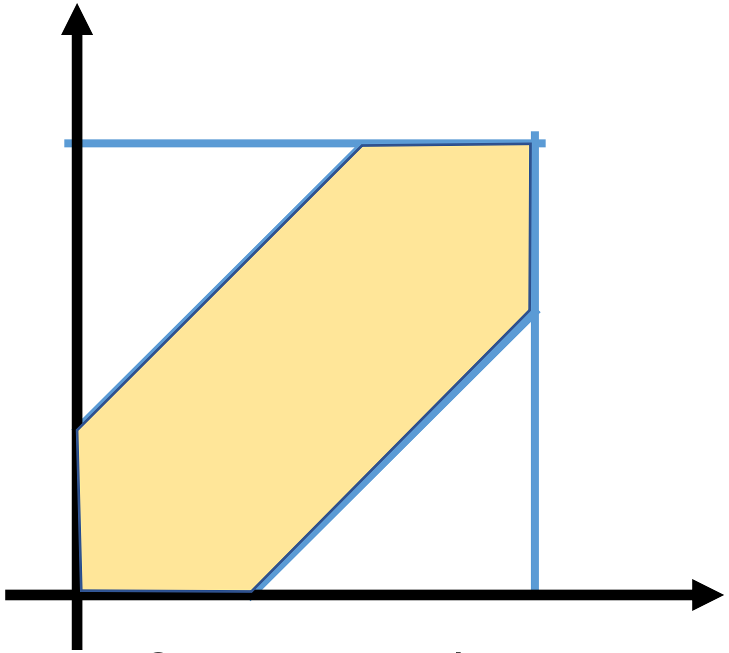
Simple logical constraints

Let x_i be a binary variable representing a selection decision

Statement	Constraint
If i selected, then j is selected	$x_i \leq x_j$
Either i is selected or j is selected, but not both	$x_i + x_j = 1$
If i is selected, then j is not selected	$x_i + x_j \leq 1$
If i is not selected, then j is not selected	$x_j \leq x_i$

Disjunctive constraints

AND



$$0 \leq x \leq 1$$

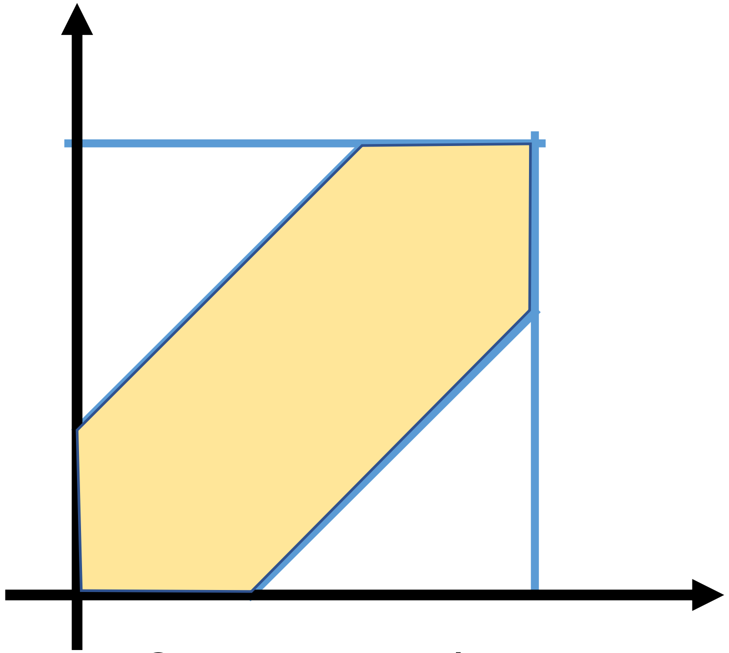
$$0 \leq y \leq 1$$

$$x + y \geq 1/3$$

$$x + y \leq 2/3$$

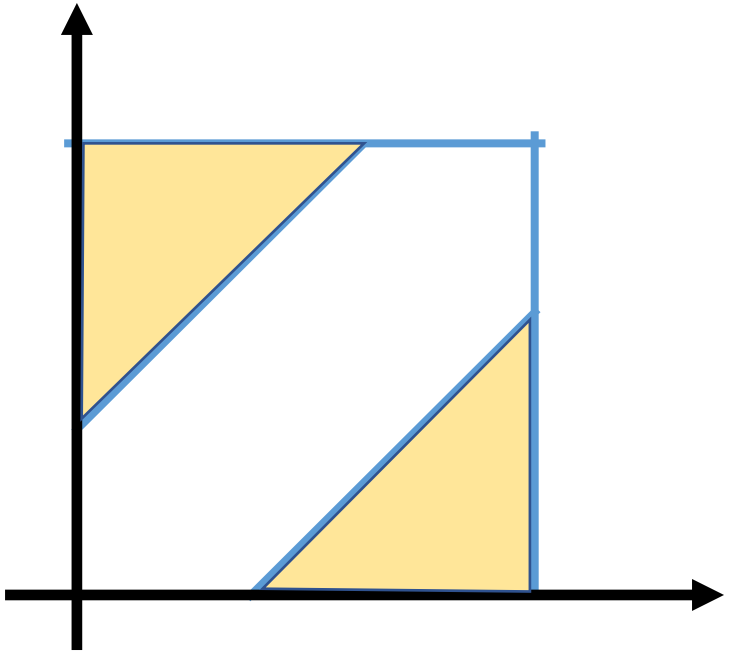
Disjunctive constraints

AND



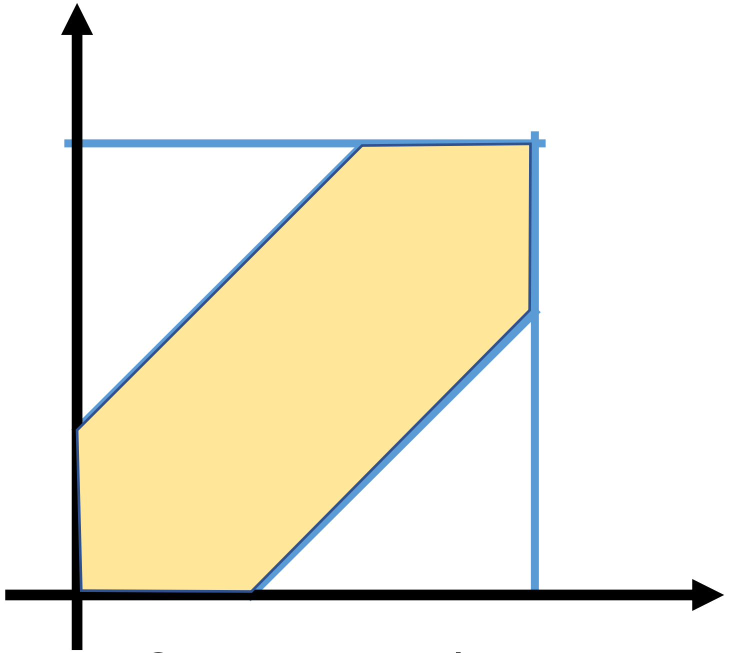
$$0 \leq x \leq 1$$
$$0 \leq y \leq 1$$
$$x + y \geq 1/3$$
$$x + y \leq 2/3$$

OR



Disjunctive constraints

AND



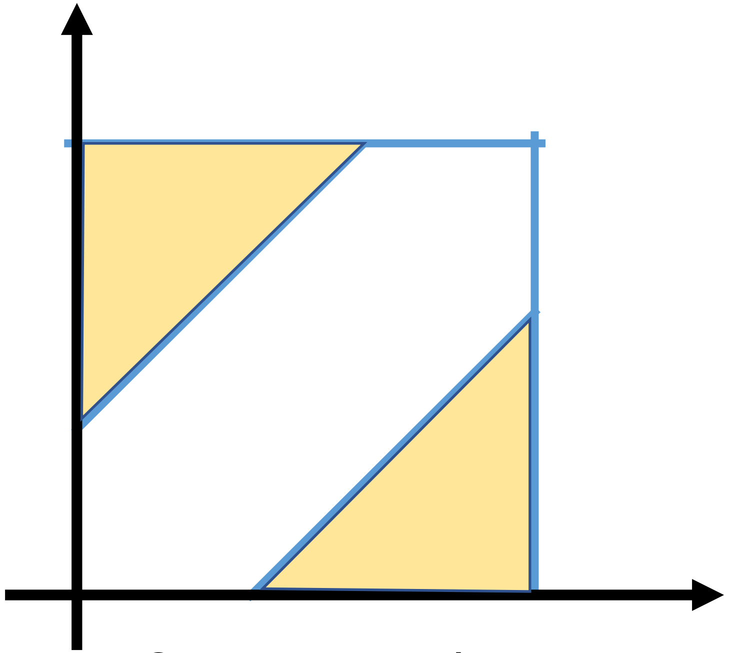
$$0 \leq x \leq 1$$

$$0 \leq y \leq 1$$

$$x + y \geq 1/3$$

$$x + y \leq 2/3$$

OR



$$0 \leq x \leq 1$$

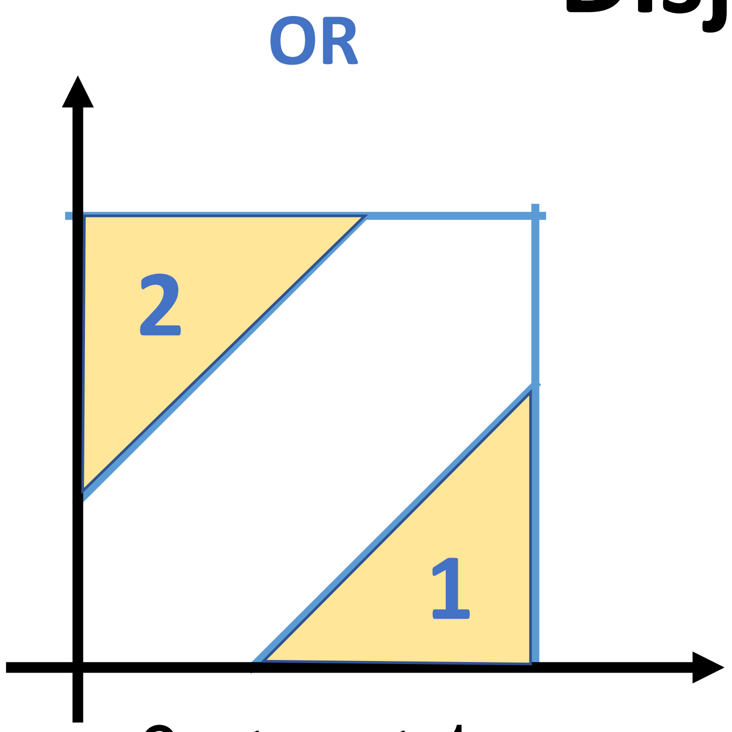
$$0 \leq y \leq 1$$

$$x + y \leq 1/3 + 10 \times w$$

$$x + y \geq 2/3 - 10 \times (1 - w)$$

$$w \in \{0, 1\}$$

Disjunctive constraints



$$0 \le x \le 1$$

$$0 \le y \le 1$$

$$x + y \le 1/3 + 10 \times w$$

$$x + y \ge 2/3 - 10 \times (1 - w)$$

$$w \in \{0, 1\}$$

$w = 0$

$x + y \ge 2/3 - 10$: always holds

$x + y \le 1/3$: region 1

$w = 1$

$x + y \ge 2/3$: region 2

$x + y \le 1/3 + 10$: always holds

Big-M method: general form

$$\sum_j a_{1j} x_j \le b_1$$

OR

$$\sum_j a_{2j} x_j \le b_2$$

$$\sum_j a_{1j} x_j \le b_1 + M_1 y$$

$$\sum_j a_{2j} x_j \le b_2 + M_2 (1 - y)$$

$$y \in \{0, 1\}$$

Big-M method: conditionals

Statement	Constraint
$z = 0 \rightarrow a^T x \leq b$	$a^T x - b \leq Mz$
$z = 0 \rightarrow a^T x \geq b$	$a^T x - b \geq mz$
$a^T x \leq b \rightarrow z = 0$	$a^T x - b \geq m(1 - z) + \varepsilon z$
$a^T x \geq b \rightarrow z = 0$	$a^T x - b \leq M(1 - z) - \varepsilon z$

Here, m and M are the lower and upper bound on $a^T x - b$, and ε is small.
The tighter the bounds, the better!

Practice problem

Lara, a cutlery manufacturing firm, produces plastic forks, knives, and spoons. Each type of cutlery is produced using a dedicated machine. These machines are rented on a weekly basis at a fixed per-week cost. Each type of cutlery requires a certain amount of plastic and labor, and can be sold at a profit; all these parameters are listed in the table below. Each week, 2000 hours of labour and 5 kg of plastic are available. How should Lara plan its production to maximize its profits? (If, say, forks are not produced, then there is no need to rent the corresponding machine and incur the fixed weekly cost.)

Cutlery item	Labor per item	Plastic per item	Profit per item	Machine rental
Knife	3 hours	5	\$2	\$2000/week
Spoon	2 hours	4	\$3	\$1500/week
Fork	6 hours	4	\$3	\$1000/week

Practice problem

Lara, a cutlery manufacturing firm, produces plastic forks, knives, and spoons. Each type of cutlery is produced using a dedicated machine. These machines are rented on a weekly basis at a fixed per-week cost. Each type of cutlery requires a certain amount of plastic and labor, and can be sold at a profit; all these parameters are listed in the table below. Each week, 2000 hours of labour and 5 kg of plastic are available. How should Lara plan its production to maximize its profits? (If, say, forks are not produced, then there is no need to rent the corresponding machine and incur the fixed weekly cost.)

Cutlery item	Labor per item	Plastic per item	Profit per item	Machine rental
Knife	3 hours	5	\$2	\$2000/week
Spoon	2 hours	4	\$3	\$1500/week
Fork	6 hours	4	\$3	\$1000/week

$$\max 2x_1 + 3x_2 + 3x_3 - 2000z_1 - 1500z_2 - 1000z_3$$

subject to

$$3x_1 + 2x_2 + 6x_3 \leq 2000$$
$$5x_1 + 4x_2 + 4x_3 \leq 5000$$
$$x_i \leq Mz_i,\ z_i \in \{0,1\},\ x_i \geq 0$$

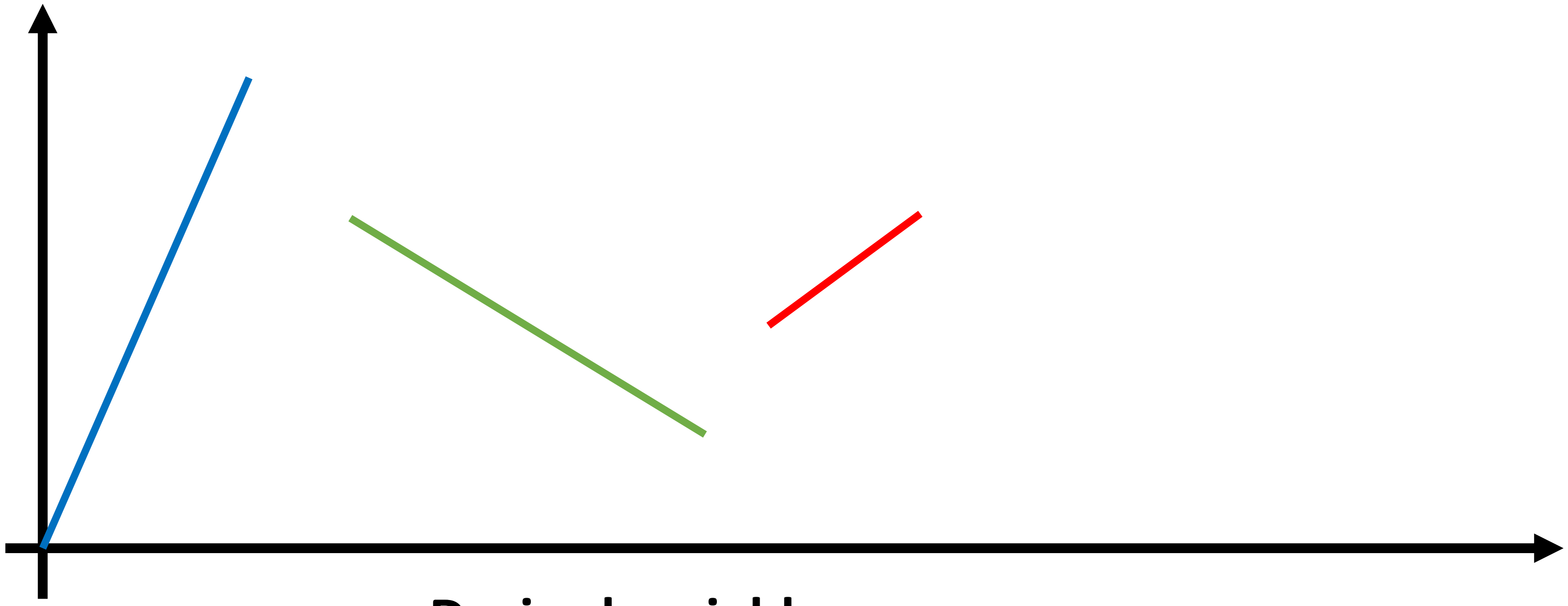
Arbitrary piecewise-linear functions

$$f(x) = \begin{cases} 2x & 0 \leq x \leq 3 \\ 9 - x & 4 \leq x \leq 7 \\ -5 + x & 8 \leq x \leq 9 \end{cases}$$

Arbitrary piecewise-linear functions

$$f(x) = \begin{cases} 2x & 0 \le x \le 3 \\ 9 - x & 4 \le x \le 7 \\ -5 + x & 8 \le x \le 9 \end{cases}$$

Arbitrary piecewise-linear functions

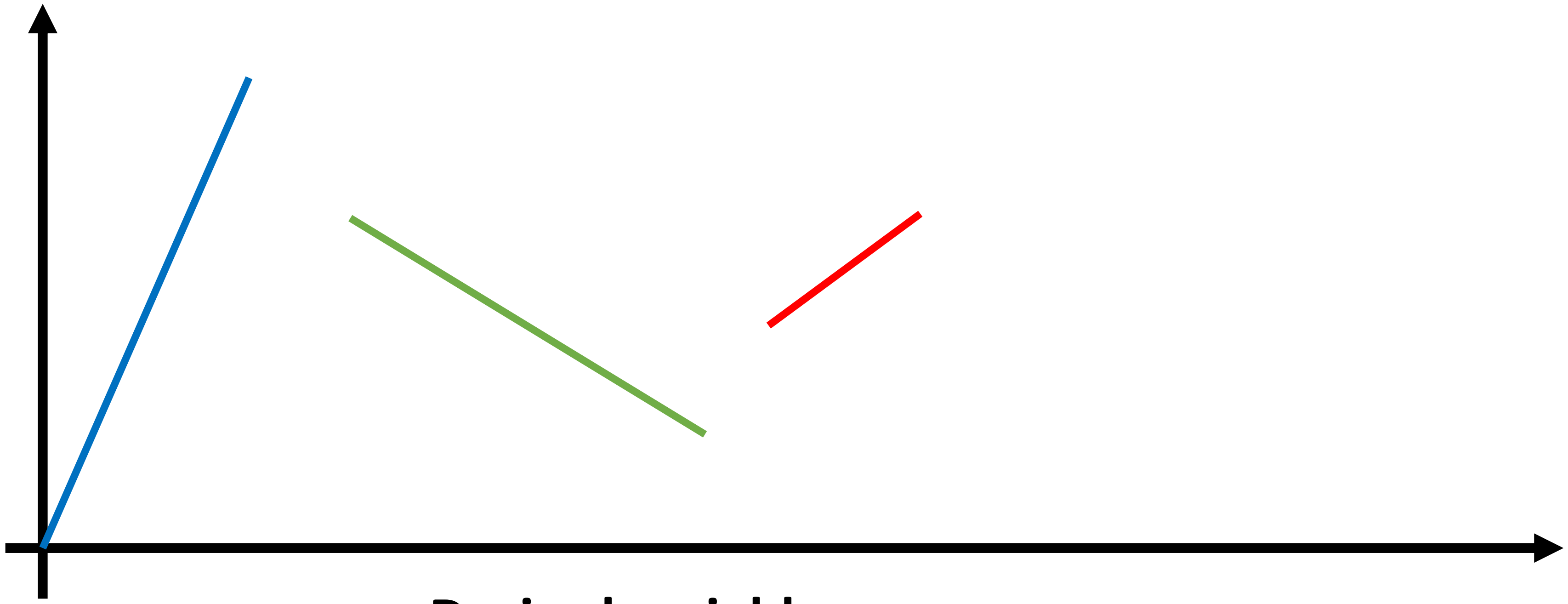


$$f(x) = \begin{cases} 2x & \underline{0 \leq x \leq 3} \\ 9 - x & \underline{4 \leq x \leq 7} \\ -5 + x & \underline{8 \leq x \leq 9} \end{cases}$$

Desired variables

	$0 \leq x \leq 3$	$4 \leq x \leq 7$	$8 \leq x \leq 9$
(w_1, x_1)	$(1, x)$	$(0, 0)$	$(0, 0)$
(w_2, x_2)	$(0, 0)$	$(1, x)$	$(0, 0)$
(w_3, x_3)	$(0, 0)$	$(0, 0)$	$(1, x)$

Arbitrary piecewise-linear functions



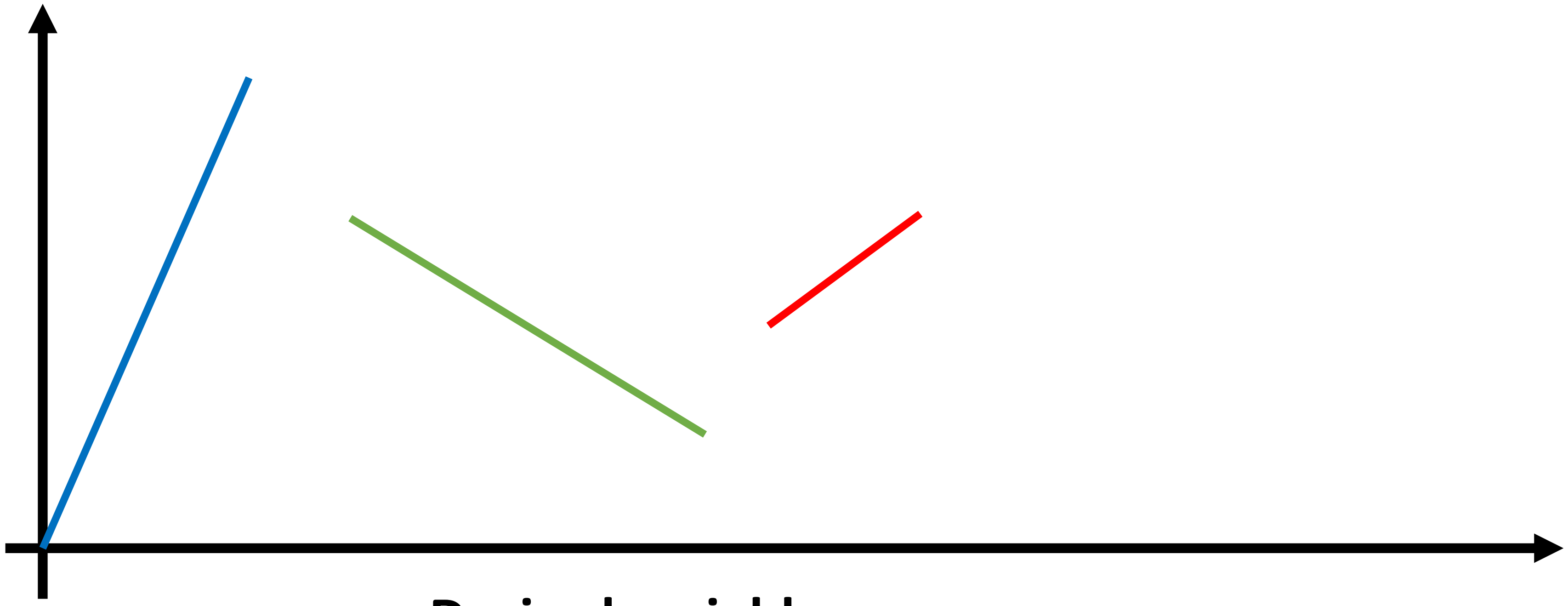
$$f(x) = \begin{cases} 2x & 0 \leq x \leq 3 \\ 9 - x & 4 \leq x \leq 7 \\ -5 + x & 8 \leq x \leq 9 \end{cases}$$

Desired variables

	$0 \leq x \leq 3$	$4 \leq x \leq 7$	$8 \leq x \leq 9$
(w_1, x_1)	$(1, x)$	$(0, 0)$	$(0, 0)$
(w_2, x_2)	$(0, 0)$	$(1, x)$	$(0, 0)$
(w_3, x_3)	$(0, 0)$	$(0, 0)$	$(1, x)$

$$f(x) = 2x_1 + (9w_2 - x_2) + (-5w_3 + x_3)$$

Arbitrary piecewise-linear functions



$$f(x) = \begin{cases} 2x & \underline{0 \le x \le 3} \\ 9 - x & \underline{4 \le x \le 7} \\ -5 + x & \underline{8 \le x \le 9} \end{cases}$$

Desired variables

	$0 \le x \le 3$	$4 \le x \le 7$	$8 \le x \le 9$
(w_1, x_1)	$(1, x)$	$(0, 0)$	$(0, 0)$
(w_2, x_2)	$(0, 0)$	$(1, x)$	$(0, 0)$
(w_3, x_3)	$(0, 0)$	$(0, 0)$	$(1, x)$

$$f(x) = 2x_1 + (9w_2 - x_2) + (-5w_3 + x_3)$$

Constraints

$$w_1, w_2, w_2 \in \{0, 1\}$$

$$w_1 + w_2 + w_3 = 1$$

$$0 \le x_1 \le 3w_1$$

$$4w_2 \le x_2 \le 7w_2$$

$$8w_3 \le x_3 \le 9w_3$$

Replacing products of variables

Replace

xy

Both Boolean

$$w$$
$$w \le x$$
$$w \le y$$
$$w \ge x + y - 1$$
$$w \in \{0, 1\}$$

x Boolean, y numeric, less than u

$$w$$
$$w \le ux$$
$$w \le y$$
$$w \ge y - u(1 - x)$$
$$w \ge 0$$

Two numerics

$$w_1 = 1/2(x + y)$$
$$w_2 = 1/2(x - y)$$

We get that

$$xy = w_1^2 - w_2^2$$

then approx. by piecewise-linear

These tricks allow us to use integer programming even when variables interact.

Specialized vs. general solvers

- We've seen how to formalize problems as integer programming ones
- But for many standard problems, specialized algorithms and solvers are available
  - These typically tend to be faster
  - Example: MiniSAT for Boolean satisfiability problem
- General integer programming solvers are useful for nonstandard problems
  - Such problems often arise while exploring
  - Integer programming is, in some sense, *declarative*
    - You describe the problem but not its solution
    - Makes it much easier to iterate
    - Perfect for prototypes

Plot twist: some IPs are LPs in disguise

- The matrix A is totally unimodular if
  - Every square submatrix of A has determinant -1, 0, or 1
- If b is an integer vector and A is totally unimodular, then for any c:
  - the value of x that maximizes $c^T x$ over the set $\{x | Ax \leq b, x \geq 0\}$ is integer
- This means that integrality constraints are redundant when A is totally unimodular and b has all integer entries
- So such integer programming problems are exceptionally tractable
- Totally unimodular matrices A often arise in **network optimization**
- Assignment problem is one such one

Recap

- We learned about integer programming
  - It’s much richer than regular linear programming
  - But can also be much tougher to solve

- Illustrated it with some applications
  - Assignment problem
  - Capital budgeting

- We found that integer programming can be used to introduce logical constraints
  - Between Boolean variables (just regular Boolean constraints)
  - Between Boolean and continuous variables (big-M method)

- Also, integer programming can be used to
  - Approximate arbitrary nonlinear functions by piecewise-linear ones
  - Remove cross-variable products

Up next

- Assignment 1 due on March 28th, 23:59

- Go over the assignment and prepare questions
  - I will answer them next time we meet
  - We will also formalize and solve more of the sample problems
  - After that, you should be able to solve all problems

- After that: variability and uncertainty
  - How queueing theory is used in service management
  - And how you can predict the behavior of the queues via simulation

- We will get back to integer and linear programming in a couple weeks

Business analytics I
Operations Analytics
Class 4
Queueing. Variability

Marat Salikhov
April 4th , 2022

Little's Law

- Inventory (work-in-process): how many units are inside the process
- Flow time: time spent by a unit within a process
- Throughput (flow rate): how many units flow through process in a unit of time
- Capacity: maximum possible throughput

- We talked a lot about flow time and throughput rate, but not about inventory
- It turns out that the inventory is linked to those two metrics via a simple relationship:

Little's Law

- Inventory (work-in-process): how many units are inside the process
- Flow time: time spent by a unit within a process
- Throughput (flow rate): how many units flow through process in a unit of time
- Capacity: maximum possible throughput

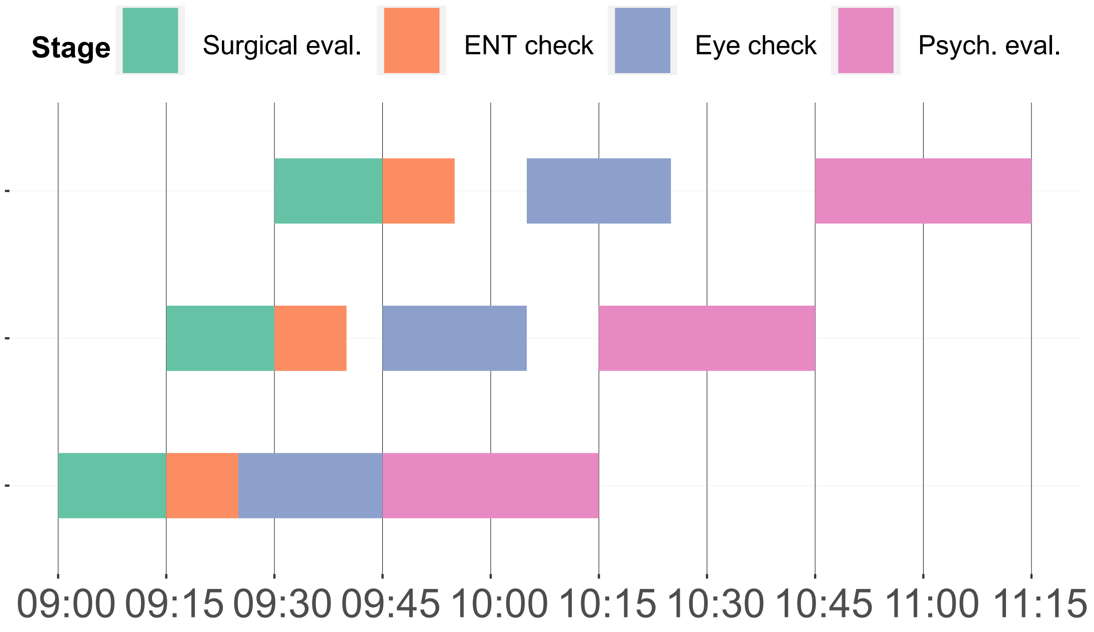
- We talked a lot about flow time and throughput rate, but not about inventory
- It turns out that the inventory is linked to those two metrics via a simple relationship:

$$\underset{\text{Inventory (WIP)}}{L} = \underset{\text{Avg. thrp. rate}}{\lambda} \times \underset{\text{Avg. flow time}}{W}$$

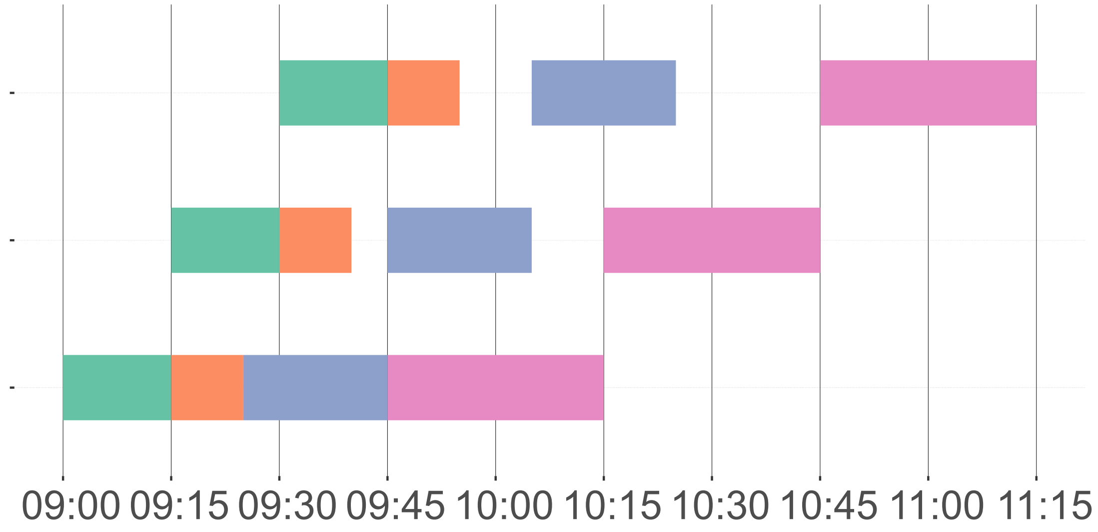
Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[units] **[units/time]** **[time]**

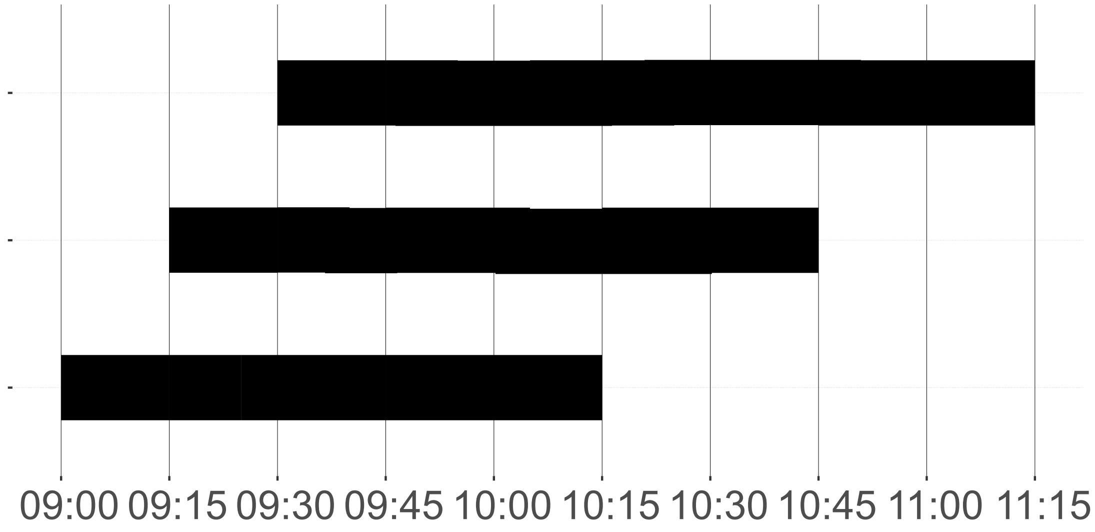
Little’s Law: intuitive argument



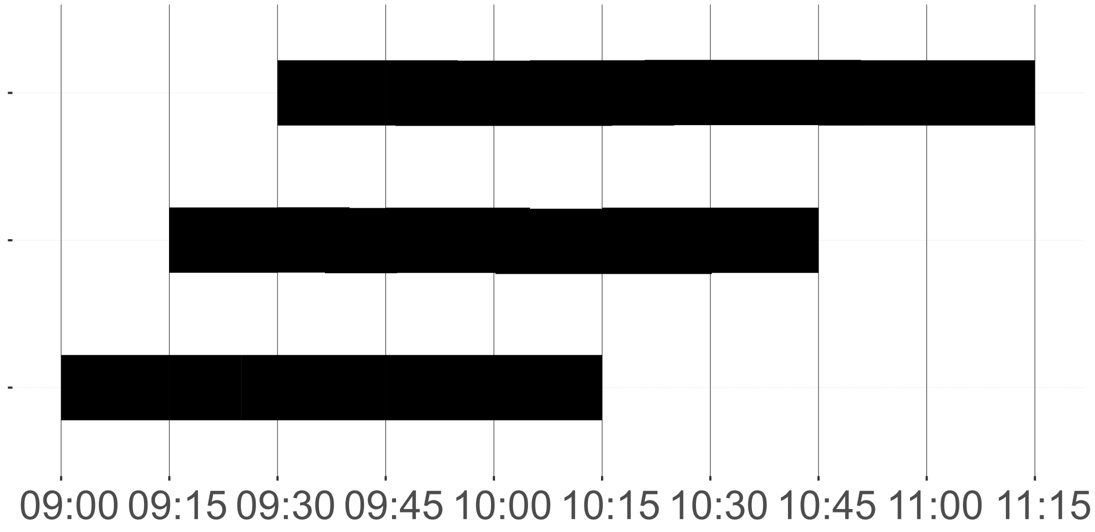
Little’s Law: intuitive argument



Little’s Law: intuitive argument



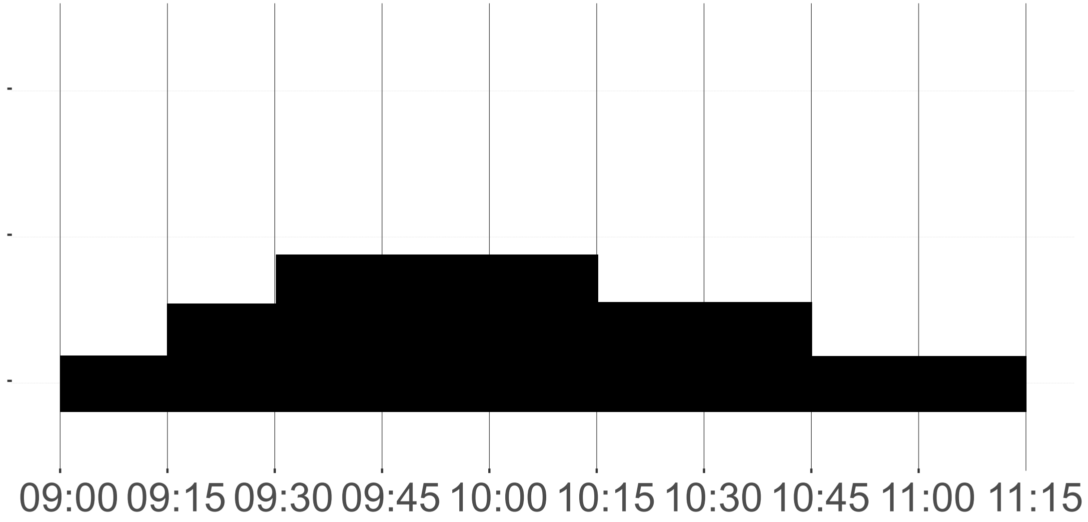
Little's Law: intuitive argument



Sum of flow times over arrivals

$$\sum_i W_i$$

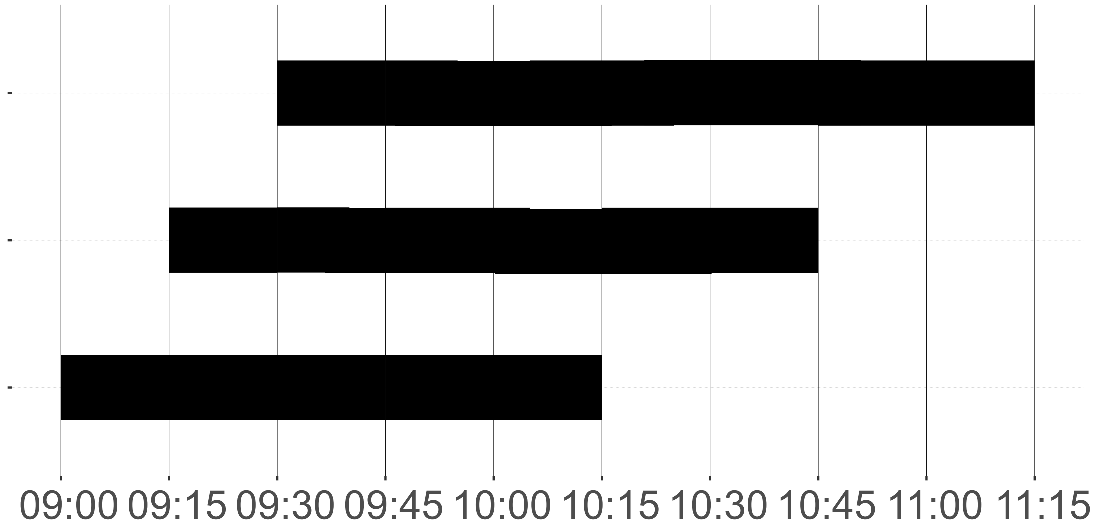
is equal to



Sum of WIP levels over time

$$\sum_t L_t$$

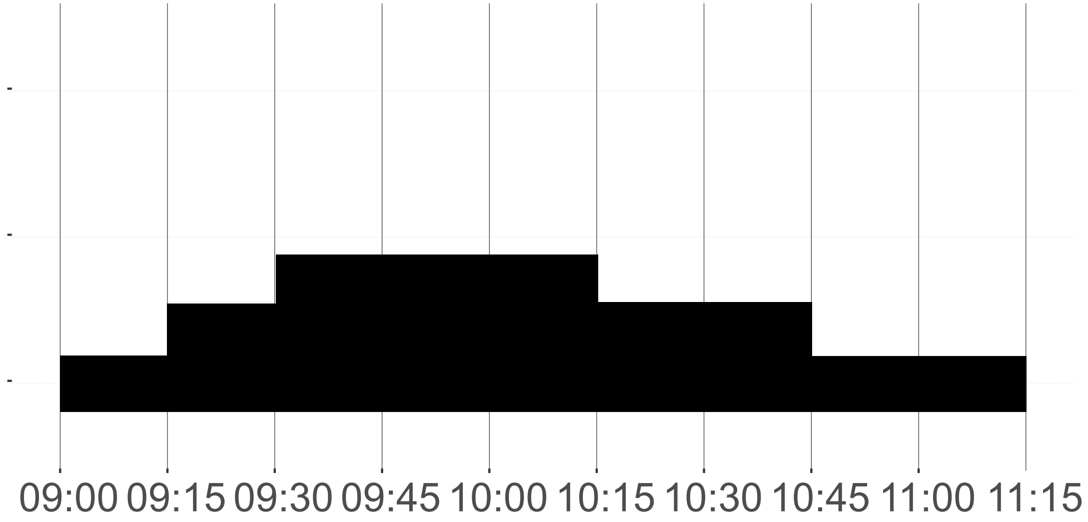
Little's Law: intuitive argument



Sum of flow times over arrivals

$$\sum_i W_i$$

is equal to



Sum of WIP levels over time

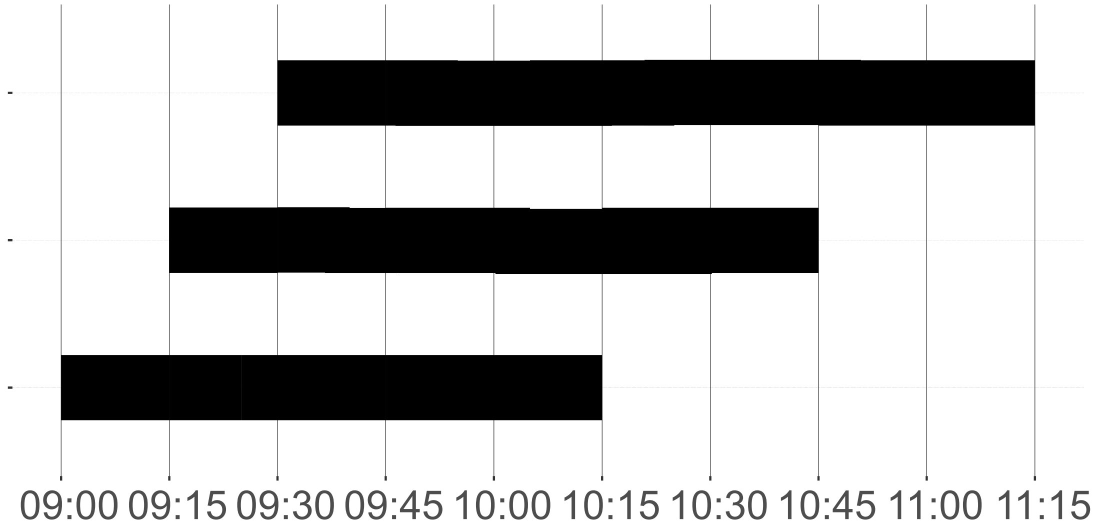
$$\sum_t L_t$$

Therefore:

$$\underbrace{\frac{1}{T}\sum_t L_t}_{\text{Avg. WIP L}} = \underbrace{\frac{N}{T}}_{\text{Arr. rate }\lambda} \times \underbrace{\frac{1}{N}\sum_i W_i}_{\text{Avg. flow time W}}$$

$$L = \lambda \times W$$

Little’s Law: intuitive argument



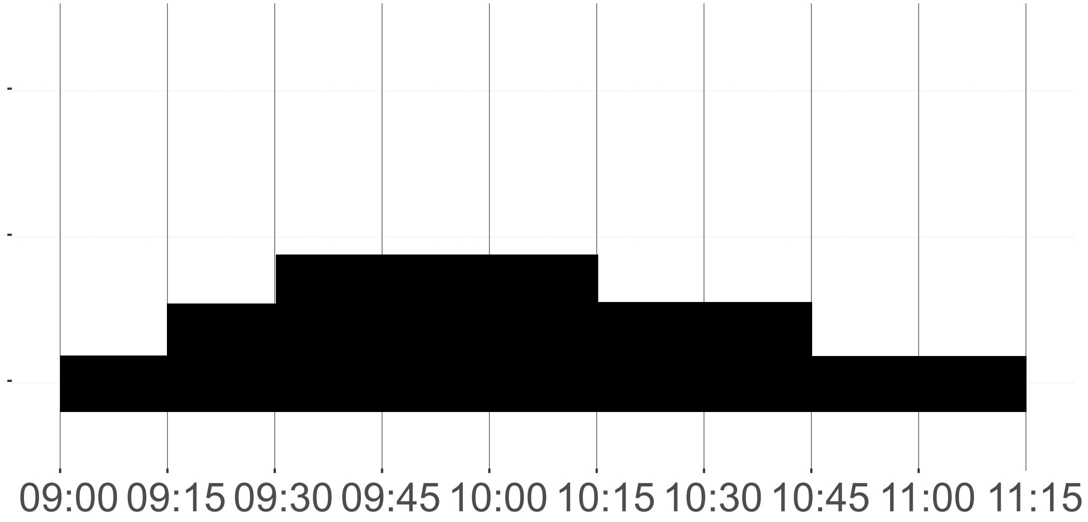
Sum of flow times over arrivals

$$\sum_i W_i$$

is equal to

Sum of WIP levels over time

$$\sum_t L_t$$



Therefore:

$$\underbrace{\frac{1}{T}\sum_t L_t}_{\text{Avg. WIP L}} = \underbrace{\frac{N}{T}}_{\text{Arr. rate } \lambda} \times \underbrace{\frac{1}{N}\sum_i W_i}_{\text{Avg. flow time W}}$$

$$L = \lambda \times W$$

General proof: quite hard

$m \to \infty$ In the last term the sum consists of a finite number (n_0) of finite terms except on the union of (n_0+1) ω-sets of probability zero Thus the sum is finite w p 1, and, since it is independent of m, the desired limit is zero w p 1 In the next to last term, $L_m \to L(\omega) < \infty$ and $\alpha/m \to 0$ w p 1 Thus

$$W(\omega) - T(\omega)L(\omega) = \lim(1/m) \sum_{j \leqq m} v(w_j + t_j - t_m) \geqq 0 \text{ w p 1}$$

If now we consider the interval $(t_{-m}(\omega), 0]$ and define L_{-m}, W_{-m}, and T_{-m} analogously to their counterparts above, e g ,

$$L_{-m} = [1/(-t_{-m})] \int_{t-m}^{0} n_s(\omega)\, ds,$$

then the symmetry of the ergodic theorems with respect to time and arguments the same as used previously yield

$$W(\omega) - T(\omega)L(\omega) = -\lim(1/m) \sum_{j \leqq -m} v(w_j + t_j - t_{-m}) \leqq 0 \text{ w p 1}$$

Therefore, $\quad W(\omega) = T(\omega)L(\omega)$ w p 1

as was to be shown

THEOREM 2 *Let*

$$L = \mathrm{E}\{n_0\}, \qquad W = \mathrm{E}\{w_0\}, \qquad T = \mathrm{E}\{\tau_0\},$$

then, under the hypotheses of Theorem 1,

$$W = TL$$

The ergodic theorems state that for almost all ω the limits (3) are the conditional expectations

$$L(\omega) = \mathrm{E}\{n_0 | \mathcal{g}_a\}, \qquad W(\omega) = \mathrm{E}\{w_0 | \mathcal{g}_b\}, \qquad T(\omega) = \mathrm{E}\{\tau_0 | \mathcal{g}_c\}$$

where $\mathcal{g}_a$, $\mathcal{g}_b$, and $\mathcal{g}_c$ are the Borel fields of invariant subsets for the corresponding processes Since the τ_r process is metrically transitive,

$$T(\omega) = T,$$

and (4) becomes $\quad W(\omega) = TL(\omega)$ w p 1

Integration over Ω gives, by definition of conditional expectation,

$$W = TL$$

as was to be shown

Little's Law: examples

$$L \qquad\qquad \lambda \qquad\qquad W$$

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[units] **[units/time]** **[time]**

1. 6000 claims per year (50 weeks). Processing time: 2 weeks. # of applications in process?

Little's Law: examples

L $\qquad\qquad\qquad$ λ $\qquad\qquad\qquad$ W

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[**units**] $\qquad$ [**units/time**] $\qquad$ **[time]**

1. 6000 claims per year (50 weeks). Processing time: 2 weeks. # of applications in process?
   λ = 6000/50 = 120 claims/week; W = 2 weeks; L = 240 claims

Little's Law: examples

$$\begin{array}{ccccc} L & & \lambda & & W \\ \text{Inventory (WIP)} & = & \text{Avg. thrp. rate} & \times & \text{Avg. flow time} \\ \text{[units]} & & \text{[units/time]} & & \text{[time]} \end{array}$$

1. 6000 claims per year (50 weeks). Processing time: 2 weeks. # of applications in process?
   λ = 6000/50 = 120 claims/week; W = 2 weeks; L = 240 claims

2. Avg. balance \$3,000. Money turned over 6 times a year.
   How many dollars you get per year?

Little's Law: examples

$$L \qquad\qquad \lambda \qquad\qquad W$$

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[units] [units/time] [time]

1. 6000 claims per year (50 weeks). Processing time: 2 weeks. # of applications in process?
   λ = 6000/50 = 120 claims/week; W = 2 weeks; L = 240 claims

2. Avg. balance $3,000. Money turned over 6 times a year.
   How many dollars you get per year?
   L = 3000$. W = 1/6. So λ = 3000 * 6 = 18,000$/year

Little's Law: examples

$$L \qquad\qquad \lambda \qquad\qquad W$$

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[**units**] [**units/time**] [**time**]

1. 6000 claims per year (50 weeks). Processing time: 2 weeks. # of applications in process?
   λ = 6000/50 = 120 claims/week; W = 2 weeks; L = 240 claims

2. Avg. balance $3,000. Money turned over 6 times a year.
   How many dollars you get per year?
   L = 3000$. W = 1/6. So λ = 3000 * 6 = 18,000$/year

3. A supermarket receives from suppliers 300 tons of fish over the course of a full year, which averages out to 25 tons per month. The average quantity of fish held in freezer storage is 16.5 tons. On average, how long does a ton of fish remain in freezer storage between the time it is received and the time it is sent to the sales department?

Little's Law: examples

$$\underset{\text{[units]}}{\overset{L}{\text{Inventory (WIP)}}} = \underset{\text{[units/time]}}{\overset{\lambda}{\text{Avg. thrp. rate}}} \times \underset{\text{[time]}}{\overset{W}{\text{Avg. flow time}}}$$

1. 6000 claims per year (50 weeks). Processing time: 2 weeks. # of applications in process?
   λ = 6000/50 = 120 claims/week; W = 2 weeks; L = 240 claims

2. Avg. balance $3,000. Money turned over 6 times a year.
   How many dollars you get per year?
   L = 3000$. W = 1/6. So λ = 3000 * 6 = 18,000$/year

3. A supermarket receives from suppliers 300 tons of fish over the course of a full year, which averages out to 25 tons per month. The average quantity of fish held in freezer storage is 16.5 tons. On average, how long does a ton of fish remain in freezer storage between the time it is received and the time it is sent to the sales department?
   L = 16.5 tons. λ = 25 tons. So W = 16.5/25 = 0.66 months.

Little's Law: examples

$$\underset{\text{[units]}}{\overset{L}{\text{Inventory (WIP)}}} = \underset{\text{[units/time]}}{\overset{\lambda}{\text{Avg. thrp. rate}}} \times \underset{\text{[time]}}{\overset{W}{\text{Avg. flow time}}}$$

A hospital emergency room (ER) is organized so that all patients register through an initial check-in process. At his/her turn, each patient is seen by a doctor and then exits the process, either with a prescription or with admission to the hospital.

Currently, 50 people per hour arrive at the ER, 10% of whom are admitted to the hospital.
On average, 30 people are waiting to be registered and 40 are registered and waiting to see a doctor.
The registration process takes, on average, 2 minutes per patient.
Among patients who receive prescriptions, average time spent with a doctor is 5 minutes.
Among those admitted to the hospital, average time is 30 minutes.

Q1: On average, how long does a patient stay in the ER?
Q2: On average, how many patients are being examined by doctors?
Q3: On average, how many patients are in the ER?

Little's Law: examples

$$\underset{\text{[units]}}{\overset{L}{\text{Inventory (WIP)}}} = \underset{\text{[units/time]}}{\overset{\lambda}{\text{Avg. thrp. rate}}} \times \underset{\text{[time]}}{\overset{W}{\text{Avg. flow time}}}$$

A hospital emergency room (ER) is organized so that all patients register through an initial check-in process. At his/her turn, each patient is seen by a doctor and then exits the process, either with a prescription or with admission to the hospital.

Currently, 50 people per hour arrive at the ER, 10% of whom are admitted to the hospital.
On average, 30 people are waiting to be registered and 40 are registered and waiting to see a doctor.
The registration process takes, on average, 2 minutes per patient.
Among patients who receive prescriptions, average time spent with a doctor is 5 minutes.
Among those admitted to the hospital, average time is 30 minutes.

Little's Law: examples

$$L \qquad\qquad \lambda \qquad\qquad W$$

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[**units**] [**units/time**] [**time**]

A hospital emergency room (ER) is organized so that all patients register through an initial check-in process. At his/her turn, each patient is seen by a doctor and then exits the process, either with a prescription or with admission to the hospital.

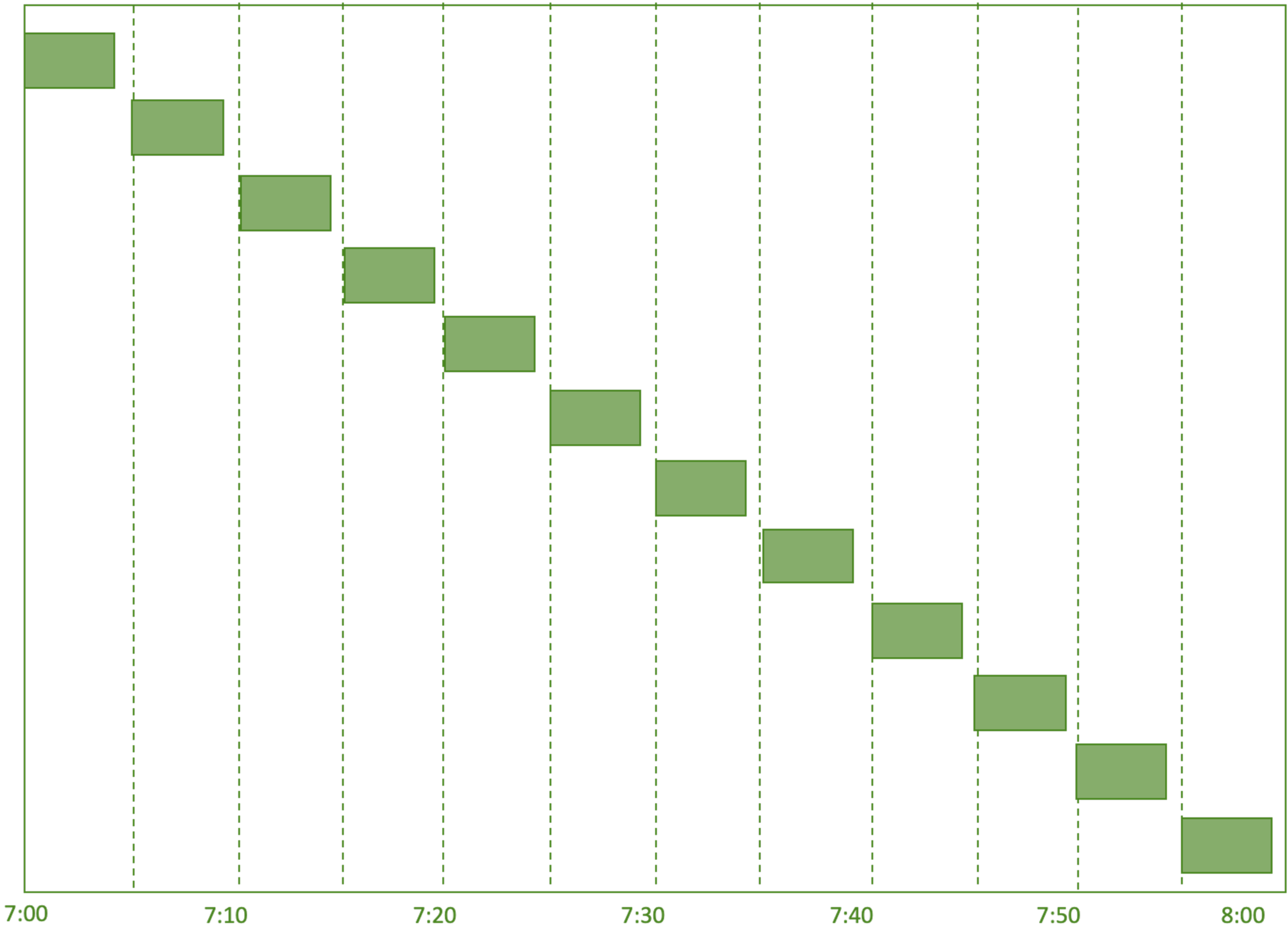
Currently, 50 people per hour arrive at the ER, 10% of whom are admitted to the hospital.
On average, 30 people are waiting to be registered and 40 are registered and waiting to see a doctor.
The registration process takes, on average, 2 minutes per patient.
Among patients who receive prescriptions, average time spent with a doctor is 5 minutes.
Among those admitted to the hospital, average time is 30 minutes.

Q1: On average, how long does a patient stay in the ER? 93.5 minutes
Q2: On average, how many patients are being examined by doctors? 6.25 patients
Q3: On average, how many patients are in the ER? 77.9 patients


21683.6
25636.8
Т289РР 178
23579
СКОРАЯ МЕДИЦИНСКАЯ
Санкт-Петербург

Service with no queues

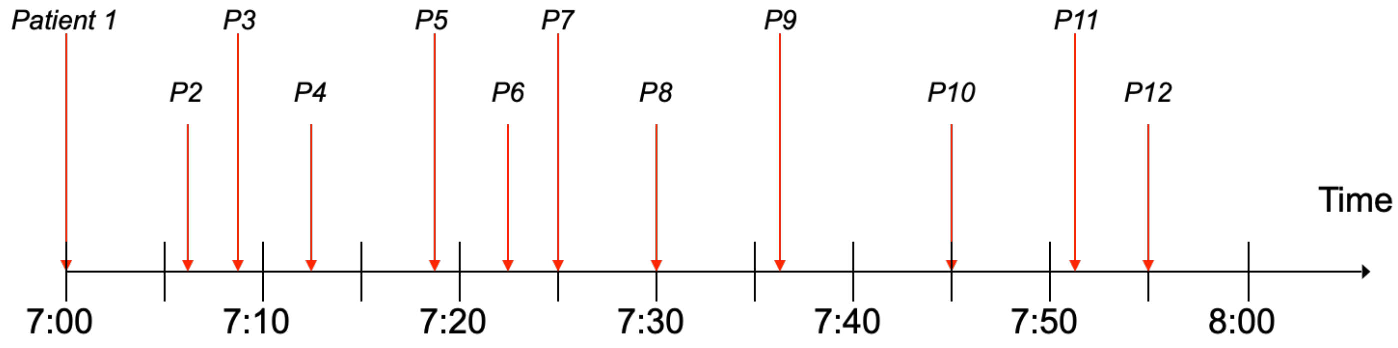
Patient	Arrival time	Interarrival time	Service time
1	7:00	0	4
2	7:05	5	4
3	7:10	5	4
4	7:15	5	4
5	7:20	5	4
6	7:25	5	4
7	7:30	5	4
8	7:35	5	4
9	7:40	5	4
10	7:45	5	4
11	7:50	5	4
12	7:55	5	4

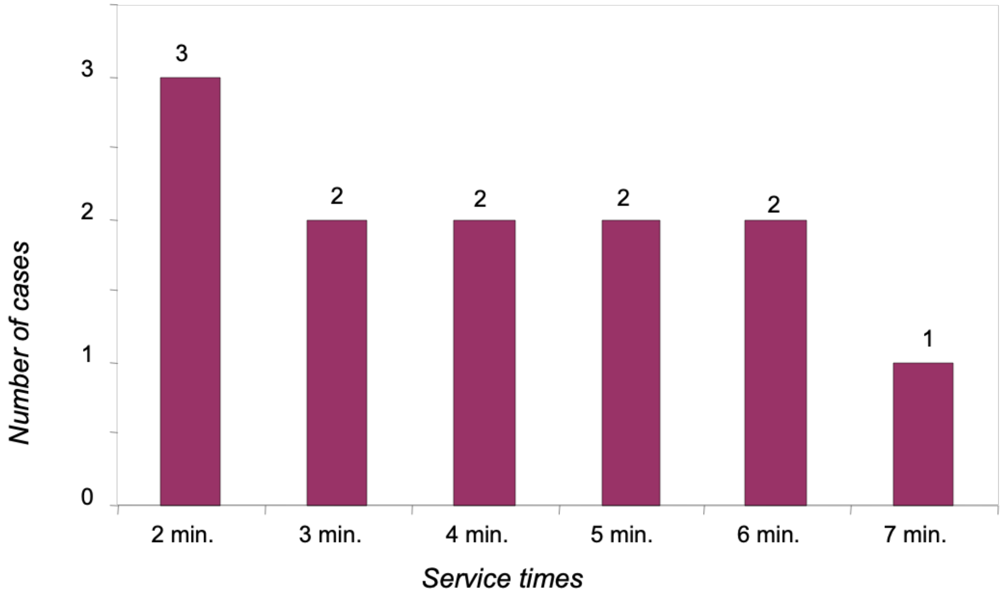


Average interarrival time is 5 minutes. Average service time is 4 minutes. No one has to wait.
Does everything look right here?

A more realistic case

Patient	Arrival time	Interarrival time	Service time
1	7:00		5
2	7:07	7	6
3	7:09	2	7
4	7:12	3	6
5	7:18	6	5
6	7:22	4	2
7	7:25	3	4
8	7:30	5	3
9	7:36	6	4
10	7:45	9	2
11	7:51	6	2
12	7:55	4	2

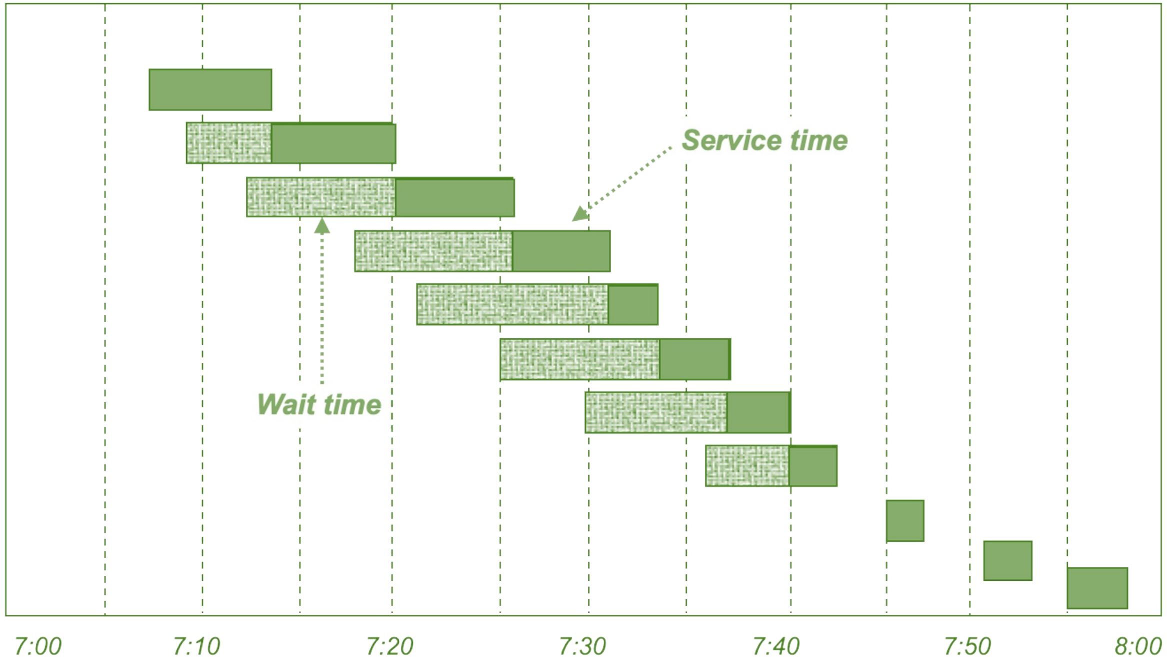


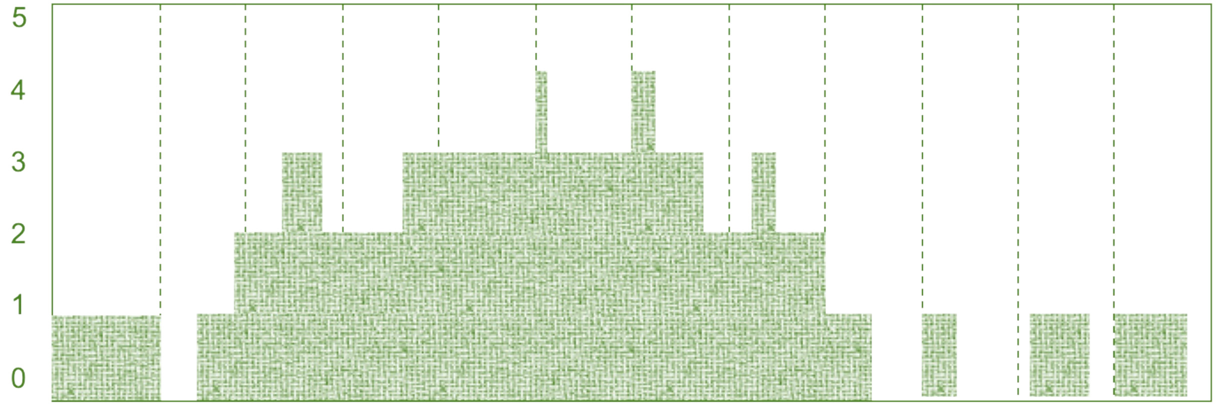


Average interarrival time is 5 minutes. Average service time is 4 minutes.
Will the service performance be the same as before?

Variability leads to waiting

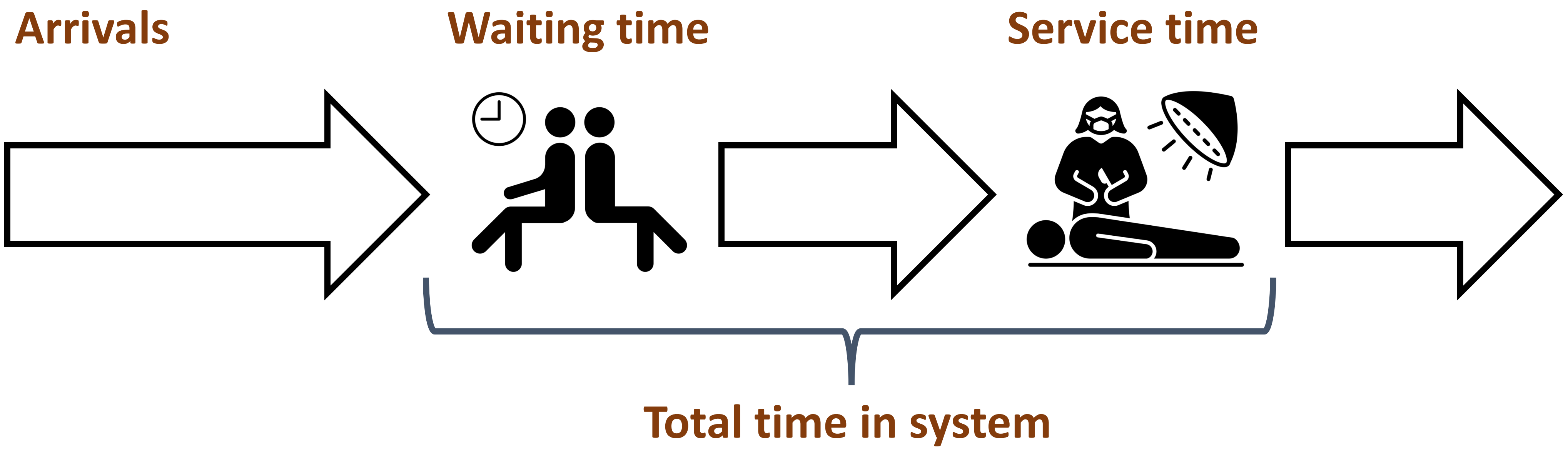
Patient	Arrival time	Interarrival time	Service time
1	7:00		5
2	7:07	7	6
3	7:09	2	7
4	7:12	3	6
5	7:18	6	5
6	7:22	4	2
7	7:25	3	4
8	7:30	5	3
9	7:36	6	4
10	7:45	9	2
11	7:51	6	2
12	7:55	4	2



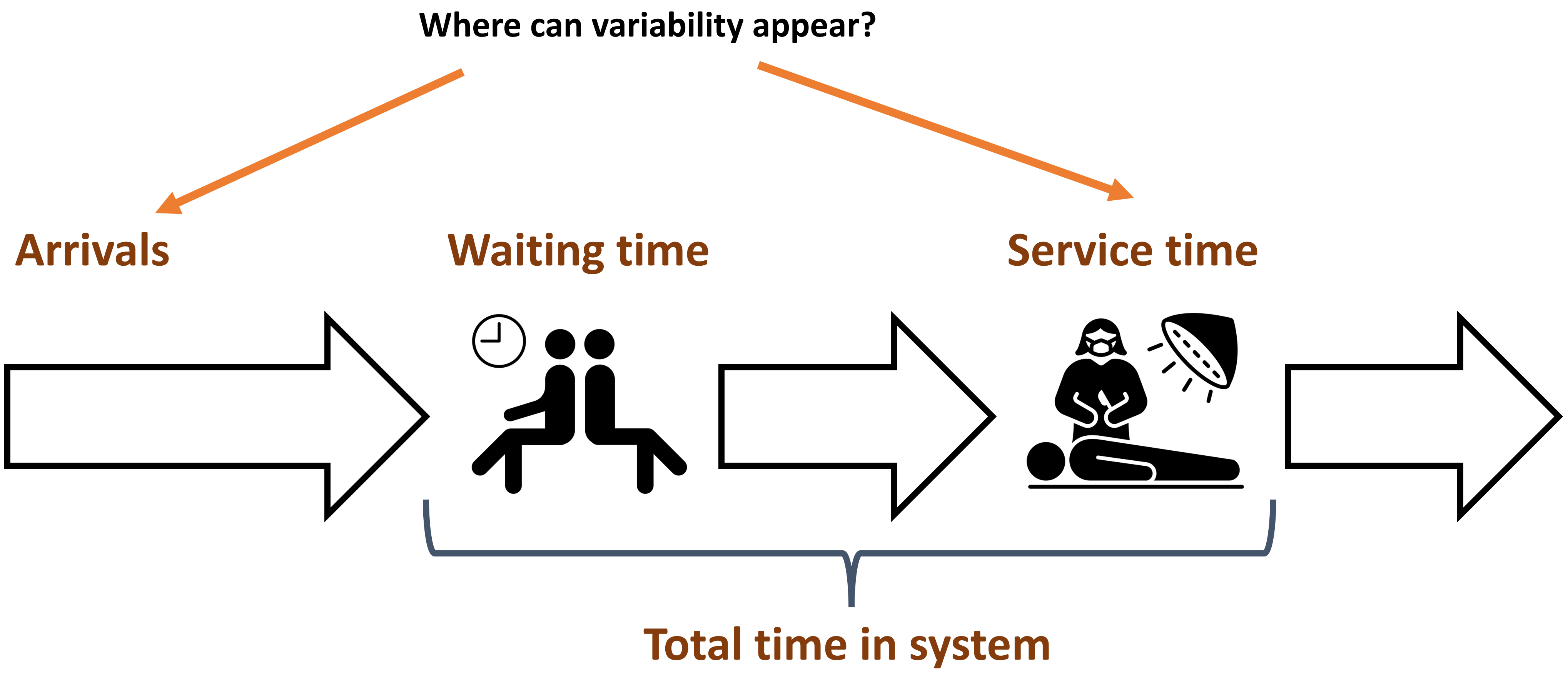


Average interarrival time is 5 minutes. Average service time is 4 minutes.
Will the service performance be the same as before?

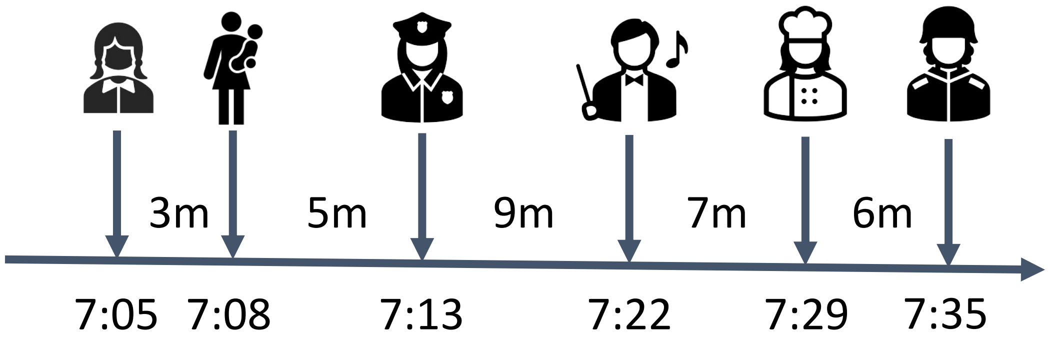
Process scheme



Process scheme



Interarrival rates statistics



Average interarrival time is **6 minutes**

AIT: $\dfrac{1}{\text{Arrival rate [cust/hr]}} = \dfrac{1}{\boldsymbol{\lambda}}$

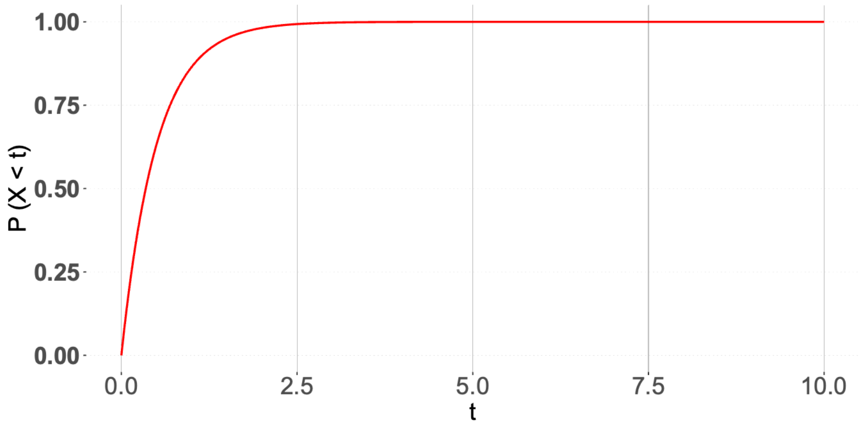
Standard deviation σ_a **is equal to** $\frac{1}{5}\sqrt{9+1+9+1+0} \approx 0.89$

Is it a lot? Normalize by AIT: $\mathrm{CV}_a = \dfrac{\boldsymbol{\sigma_a}}{\mathbf{AIT}} \approx 0.15$

Exponential distribution

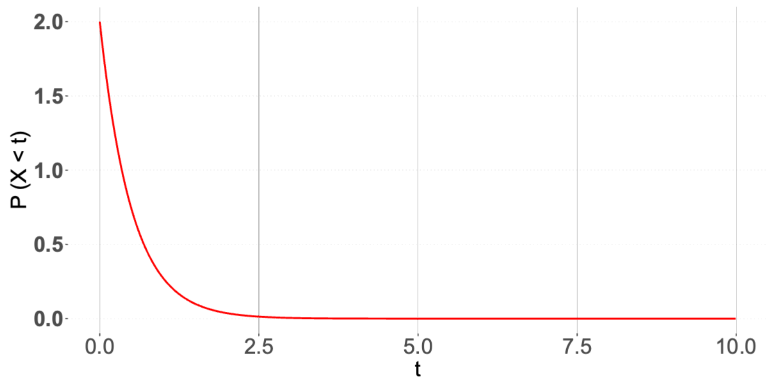
CDF

$$\mathbb{P}(X < t) = 1 - \exp(-t/\mu)$$



PDF

$$\frac{d\mathbb{P}(X<t)}{dt} = \frac{1}{\mu}\exp(-t/\mu)$$



If X is an exponential variable with $\mathbb{E}X = \mu$, then $\mathbb{V}X = \mu^2$ and so the coefficient of variation is equal to 1.

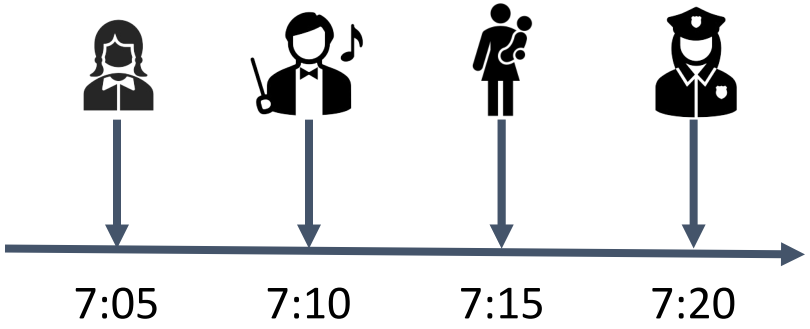
If the arrival times between events are distributed exponentially, then the arrival process is called a *Poisson* process.

If X and Y are independent exponential random variables with means μ_X and μ_Y then $\min(X, Y)$ is distributed exponentially with mean $\mu_X + \mu_Y$. So if you merge two independent Poisson processes, you get another Poisson process.

Arrivals with different CVs

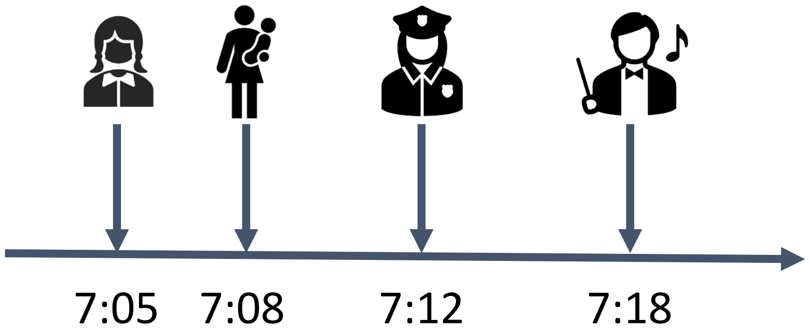
A process with CV = 0

Perfectly scheduled arrivals.
Example: **subway in Singapore**



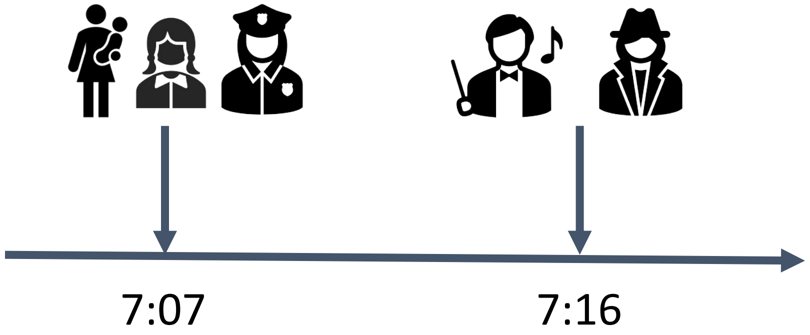
A process with CV = 1

No or weak relationship between arrivals.
Example: **calls to a call center**
Poisson process has CV = 1
(interarrival times exponential and independent)



A process with CV >> 1

Batched arrivals: **coffee orders at a cafe**



Service rate statistics

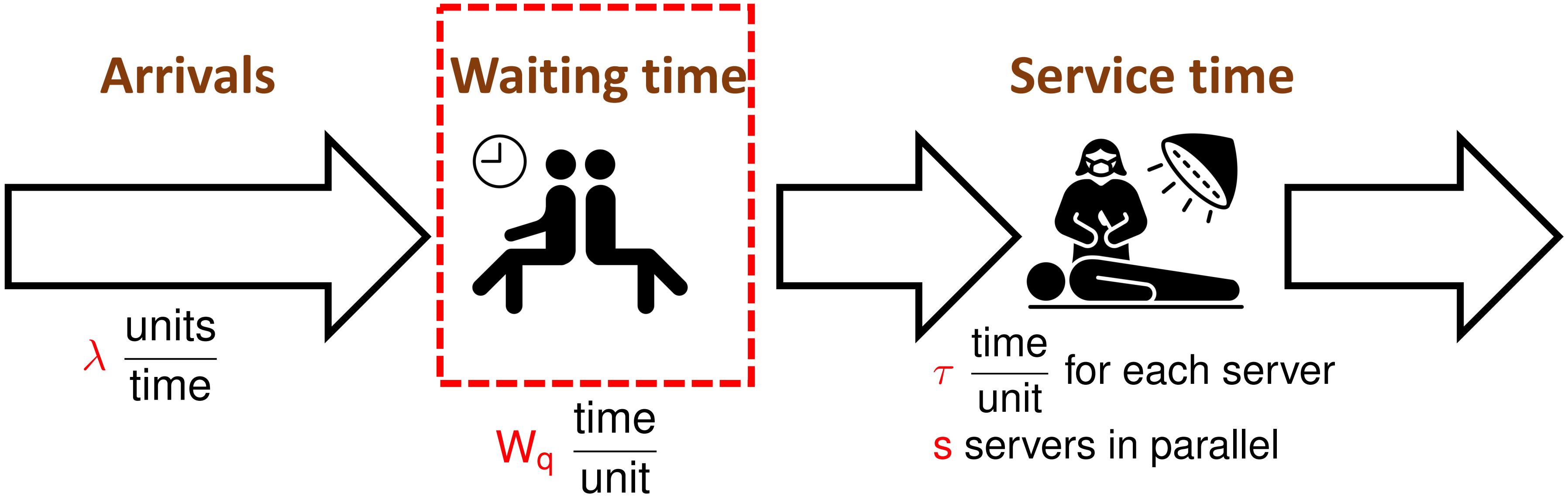
Patient	Arrival Time	Inter-Arrival Time	Service Time
1	7:00		5
2	7:07	7	6
3	7:09	2	7
4	7:12	3	6
5	7:18	6	5
6	7:22	4	2
7	7:25	3	4
8	7:30	5	3
9	7:36	6	4
10	7:45	9	2
11	7:51	6	2
12	7:55	4	2

Average service time τ = 4 min

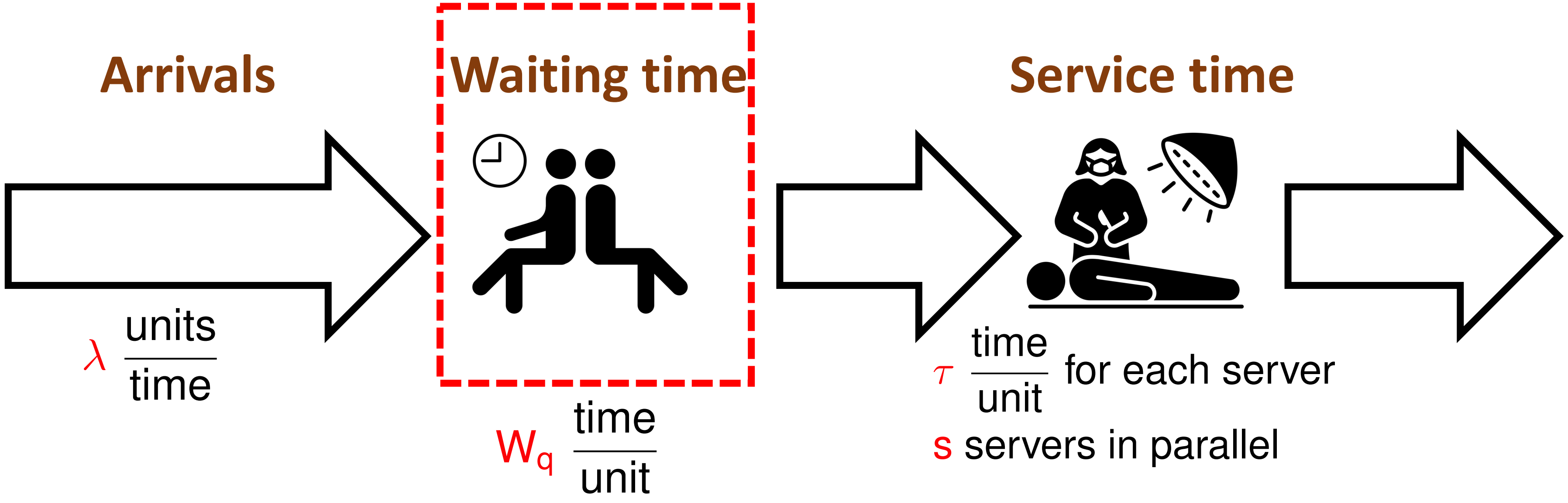
Std.dev. of service time σ_s = 1.73 min

Coefficient of variation $CV_s = 1.73/4 = 0.4325$

How long will you wait in a queue?



How long will you wait in a queue?

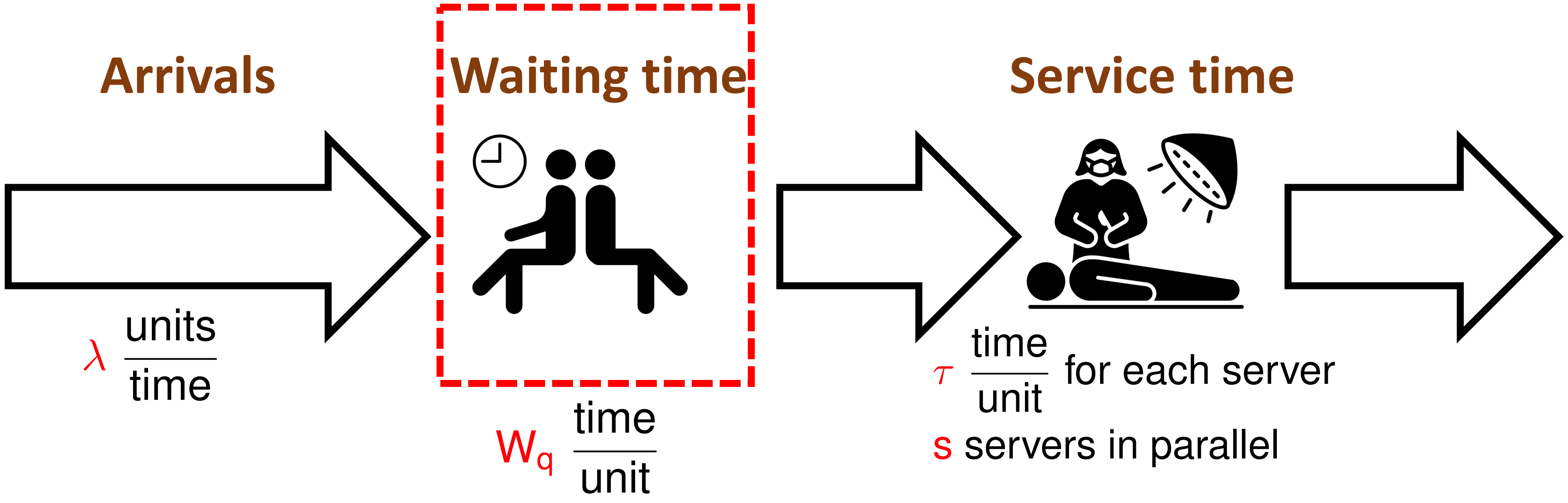


Avg. waiting time: $$W_q \approx \frac{\tau}{s} \times \frac{\rho^{\sqrt{2(s+1)}-1}}{1-\rho} \times \frac{CV_a^2 + CV_s^2}{2}$$

Service rate per server ↓ Utilization ↓ Variability ↓

Utilization: $\rho = \frac{\lambda\tau}{s}$

Sakasegawa's approximation



Avg. waiting time: $$W_q \approx \frac{\tau}{s} \times \frac{\rho^{\sqrt{2(s+1)}-1}}{1-\rho} \times \frac{CV_a^2 + CV_s^2}{2}$$

Service rate per server / Utilization / Variability

Utilization: $\rho = \frac{\lambda\tau}{s}$

This formula is a numerical approximation.
Precise results are hard for general interarrival and service times.

An approximation formulaL q ≃α·ρ β /(1-ρ)

Hirotaka Sakasegawa

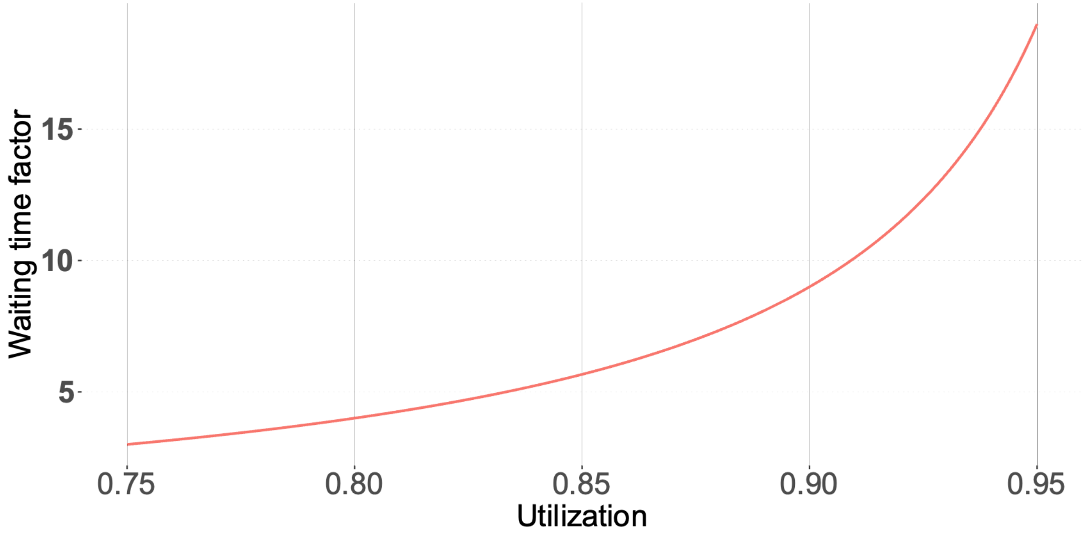
Annals of the Institute of Statistical Mathematics, 1977, vol. 29, issue 1, 67-75

High utilization hurts disproportionately

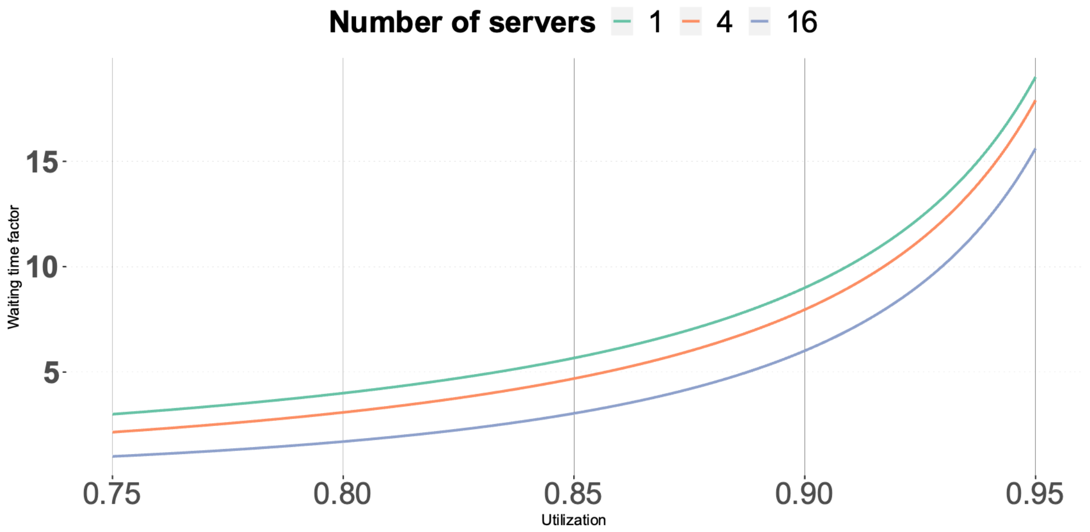
Waiting time factor for utilization

$$\frac{\rho^{\sqrt{2(s+1)}-1}}{1-\rho}$$

Single server:



Multiple servers:

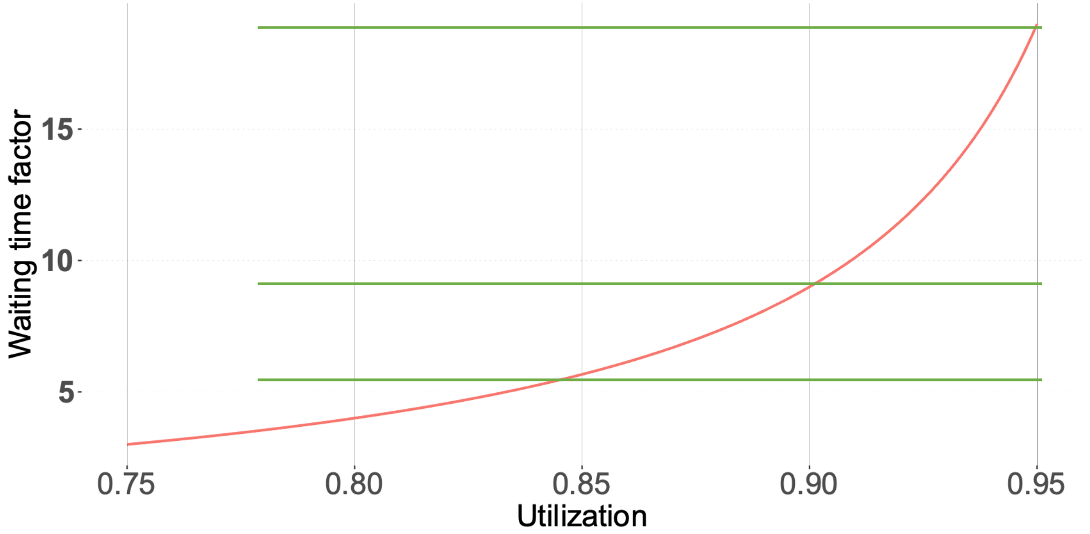


High utilization hurts disproportionately

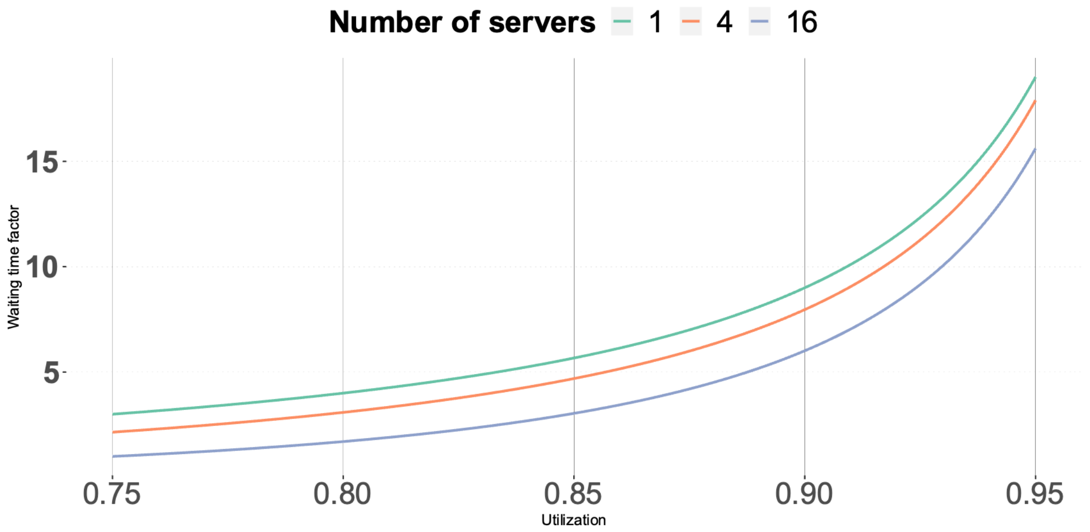
Waiting time factor for utilization

$$\frac{\rho^{\sqrt{2(s+1)}-1}}{1-\rho}$$

Single server:

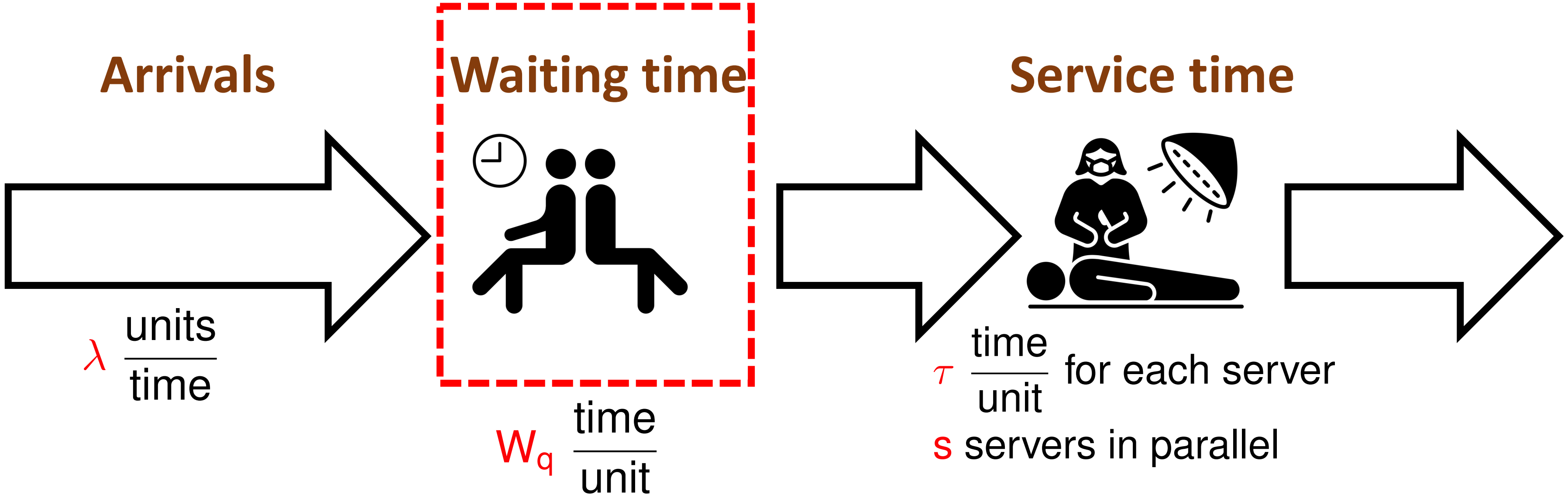


Multiple servers:

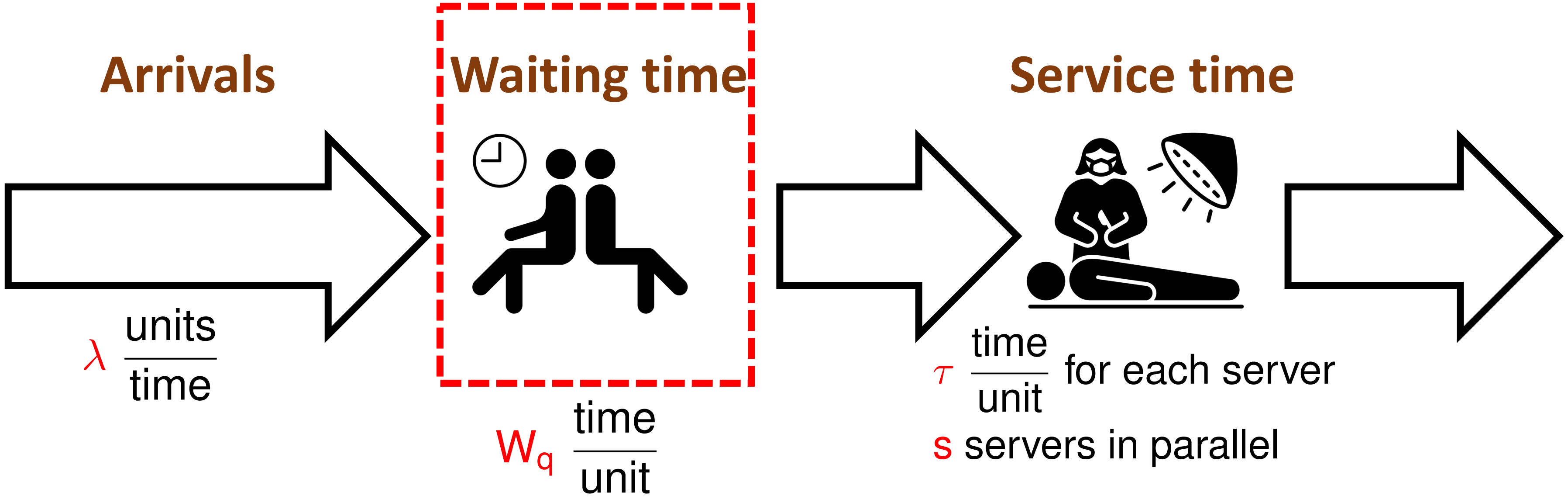


100% utilization might be quite bad for customers

Average length of the queue?



Average length of the queue? Little's Law



Avg. queue length: $L_q = \lambda \times W_q$

Little's law: Arrival rate $\times$ Waiting time = Inventory also works for queues.

Example: call center

There is one operator who receives 10.8 calls per hour on average. The operator can handle 12 calls per hour. Suppose that the coefficient of variation of the arrival rates is 0.99 and of the service rates is 0.7. What is the average wait before the service begins? What is the average total time in system?

Utilization:

Waiting time:

Total time in system:

Example: call center

There is one operator who receives 10.8 calls per hour on average. The operator can handle 12 calls per hour. Suppose that the coefficient of variation of the arrival rates is 0.99 and of the service rates is 0.7. What is the average wait before the service begins? What is the average total time in system?

Utilization:

$$\rho = \frac{\lambda\tau}{\mathrm{s}} = \frac{10.8}{12} = 90\%$$

Waiting time:

$$\mathrm{W_q} \approx \frac{\tau}{\mathrm{s}} \times \frac{\rho}{1-\rho} \times \frac{\mathrm{CV_a^2} + \mathrm{CV_s^2}}{2} \approx 33 \text{ min}$$

Total time in system:

$$\mathrm{W} = \mathrm{W_q} + \tau \approx 38 \text{ min}$$

Example: call center

Suppose that you want customers to wait no longer than 45 seconds before being served.
How many additional operators do you need to hire?

Example: call center

Suppose that you want customers to wait no longer than 45 seconds before being served. How many additional operators do you need to hire?

$s = 2$

$\rho = 45\%$

$$W_q \approx 2.5 \times \frac{0.45^{\sqrt{6}-1}}{0.55} \times 0.74 = 1.06 \text{ min}$$

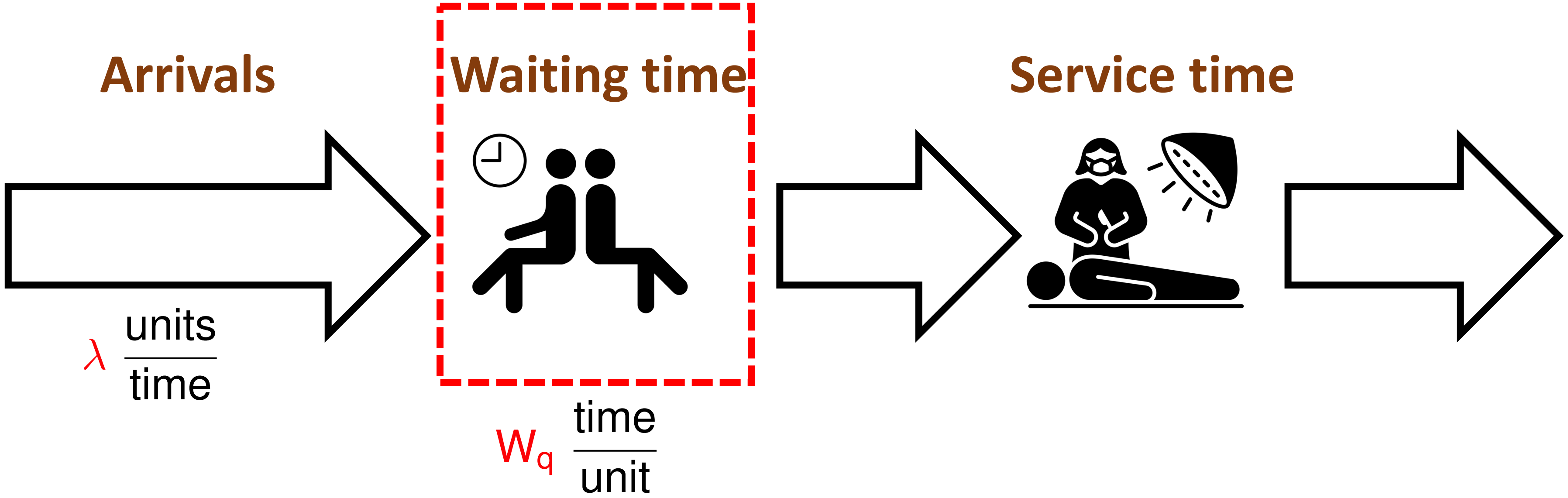
$s = 3$

$\rho = 30\%$

$$W_q \approx 1.66 \times \frac{0.3^{\sqrt{8}-1}}{0.7} \times 0.74 = 0.19 \text{ min}$$

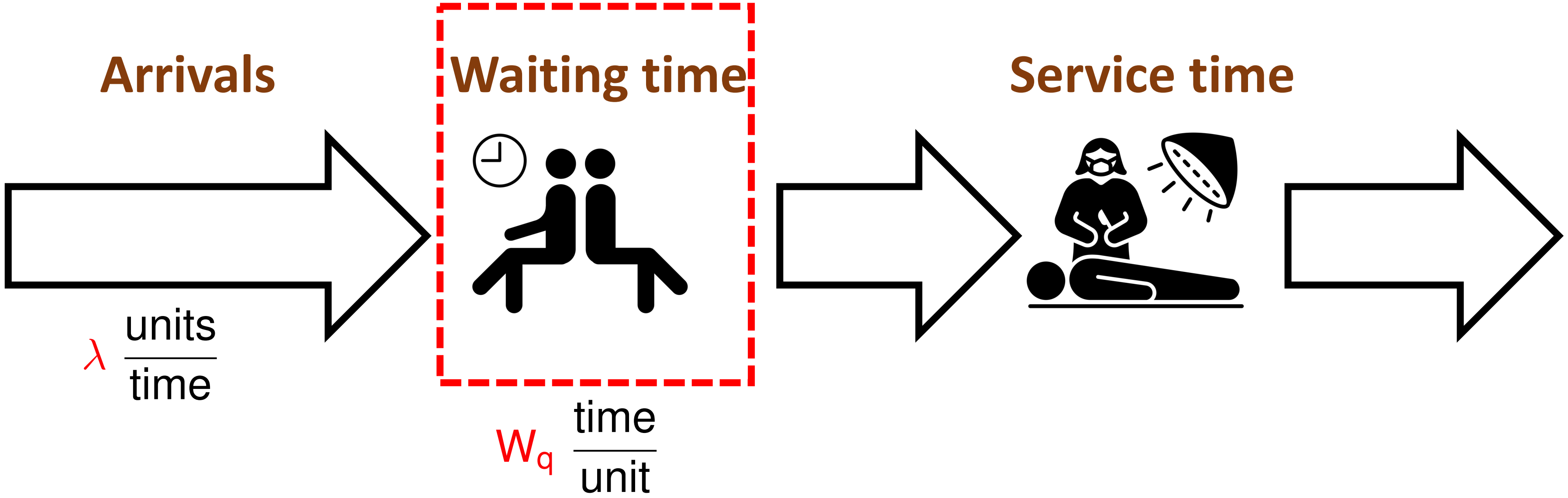
Two more operators are needed.

How to reduce waiting times?



Avg. waiting time: $$W_q \approx \frac{\tau}{s} \times \frac{\rho^{\sqrt{2(s+1)}-1}}{1-\rho} \times \frac{CV_a^2 + CV_s^2}{2}$$

How to reduce waiting times?

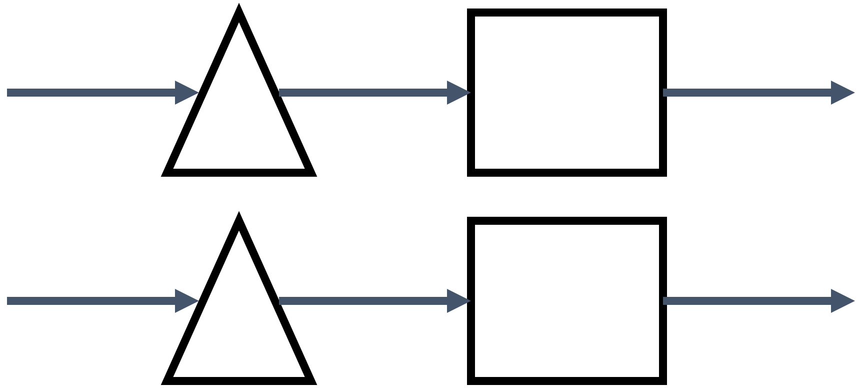


Avg. waiting time: $$W_q \approx \frac{\tau}{s} \times \frac{\rho^{\sqrt{2(s+1)}-1}}{1-\rho} \times \frac{CV_a^2 + CV_s^2}{2}$$

1. Speed up service time
2. Add servers: reduction in utilization and speed up service
3. Reduce variability of service times and arrivals

Pooling

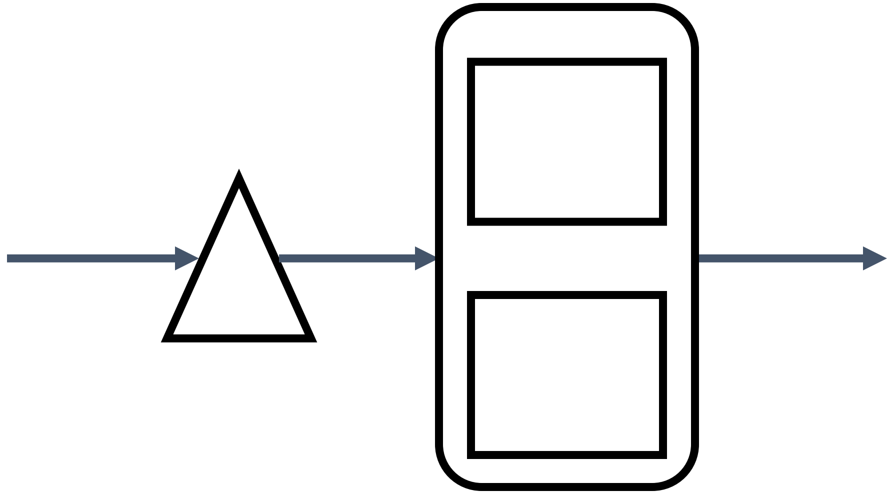
Independent resources (s = 1)



Interarrival time: 4 minutes
Service time: 3 minutes
Coefficients of variation: 1 for both
Capacity utilization: ¾ =75%

$$W_q = 3 \times \frac{0.75}{1-0.75} \times \frac{1+1}{2} = 9 \text{ min}$$

Pooled resources (s = 2)



$$W_q = \frac{3}{2} \times \frac{0.75^{\sqrt{6}-1}}{1-0.75} \times \frac{1+1}{2} = 3.95 \text{ min}$$

The new time is less than half of the old one!

Pooling: advantages and drawbacks

- Benefits:
  - Reduced customer waiting time without investing in capacity
  - Or: reduced capacity without decreasing responsiveness
- Drawbacks:
  - Pooling might require cross-training and using flexible resources
  - Process variability (CVs) might actually increase
  - Might require additional setups
  - Space requirements
  - No continuous customer followup
    - Lack of “ownership” over one’s work

Simulation

- So far we've been looking at simple models
  - And used approximations to understand their behaviour

- How can we get a better understanding of how models behave?

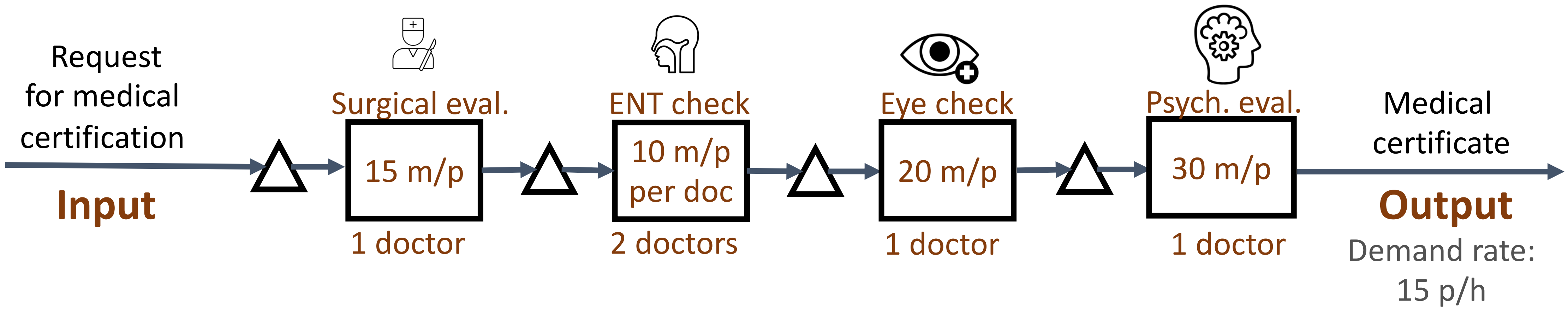
- One alternative: study the math
  - Doesn't give you answers to everything
  - Might get very complicated

- Another alternative
  - Simulation: build a program that replicates the system's behaviour
  - Still need to make sure that it's estimating the right value
  - But much easier than using analytical solutions

- We will go through stochastic simulation in our next two sessions

Project ideas

Project 1: Processes with variability



- In the “classical” process analysis all the processing times are assumed to be deterministic. In this case, buffers are not important.

- What if they are random? What happens to the overall productivity of the process, and do buffers matter?

- Formulate an optimization problem that could represent process analysis with random processing times

- Solve it for some interesting example. Show how it behaves when e. g. variability goes up.

Project 2: Optimal product line design

Which ones to pick given preferences?

- Read through McBride and Zufryden paper
  - Will be uploaded to my.nes
- Implement their integer programming problem
- Illustrate how it works on a ”made-up” dataset
- Explore how those product lines depend on problem parameters

Project idea 3: Interpretable rule design

https://arxiv.org/pdf/2103.11251.pdf

- Machine learning: often produces inscrutable models

- Not OK when it affects people's lives seriously
  - E. g. cancer screening, or deciding whether a person can get on bail

- A good model should predict well and be understandable to those affected

- Example of such models: scoring systems.
  - E. g. predicting violent crime
    - -10 points for petty theft, 9 for weapon use, -9 for employment
    - Predicting the outcome: take a sum of all points and see if they're greater than 0

Project idea 3: Interpretable rule design

https://arxiv.org/pdf/1306.5860.pdf

Implement this paper

sparse models. The objective is minimized by a MIP with $N + 3P$ variables and $2N + 4P$ constraints:

$$
\begin{aligned}
\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\lambda}} \; & \frac{1}{N}\sum_{i=1}^{N}\alpha_i + C_0\sum_{j=1}^{P}\beta_j + C_1\sum_{j=1}^{P}\gamma_j \\
-M\alpha_i + \epsilon \le y_i\mathbf{x}_i^T\boldsymbol{\lambda} & \le M(1-\alpha_i) + \epsilon \qquad i = 1\dots N \\
-\Lambda\beta_j \le \quad \lambda_j \quad & \le \Lambda\beta_j \qquad j = 1\dots P \\
-\gamma_j \le \quad \lambda_j \quad & \le \gamma_j \qquad j = 1\dots P \\
\boldsymbol{\lambda} & \in \mathcal{L} \\
\alpha_i & \in \{0,1\} \qquad i = 1\dots N \\
\beta_j \in \{0,1\}, \gamma_j & \in \mathbb{R}_+ \qquad j = 1\dots P
\end{aligned}
$$

where $\alpha_i = \mathbb{1}[y_i \neq \hat{y}_i], \beta_j = \mathbb{1}[\lambda_j \neq 0]$, and $\gamma_j = |\lambda_j|$. All feasible $\boldsymbol{\lambda}$ belong to $\{\boldsymbol{\lambda} \in \mathcal{L} : |\lambda_j| \le \Lambda \; \forall j\}$. By default, we set $\Lambda = 100$ and $\mathcal{L} = \mathbb{Z}^P$, although we often further restrict the coefficients to have only one significant digit. Lastly, ϵ and M are scalars used in if-then constraints; we set $\epsilon = 0.1, M = \Lambda \cdot \max_{i,j}|x_{ij}|$.

- Come up with a dataset (maybe use some of theirs)
- Implement their MIP problem in cvxpy
- Comment on some of the properties of this method
  - E. g. is it very sensitive to outliers in the data?

Up next

- Assignment 1 due on April 18th. Also, submit your team roster to me.

- More about simulation and random variables

- Seminar: how to simulate queues?
  - Simulating random variables
  - Queue as a data structure.
  - Basics of discrete event simulation

- Building up to:
  - Stochastic optimization: optimal decisions under uncertainty
  - How does this affect the firms' business models?

Business analytics I
Operations Analytics
Class 5
Markovian modeling. Newsvendor

Marat Salikhov

April 11th , 2022

Little's Law example

$$L \qquad \lambda \qquad W$$

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[**units**] [**units/time**] [**time**]

A hospital emergency room (ER) is organized so that all patients register through an initial check-in process. At his/her turn, each patient is seen by a doctor and then exits the process, either with a prescription or with admission to the hospital.

Currently, 50 people per hour arrive at the ER, 10% of whom are admitted to the hospital.
On average, 30 people are waiting to be registered and 40 are registered and waiting to see a doctor.
The registration process takes, on average, 2 minutes per patient.
Among patients who receive prescriptions, average time spent with a doctor is 5 minutes.
Among those admitted to the hospital, average time is 30 minutes.

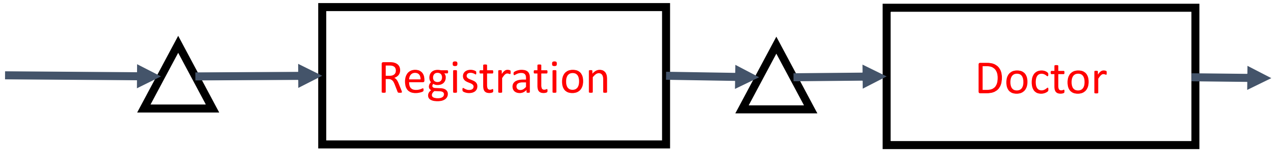
Q1: On average, how long does a patient stay in the ER? 93.5 minutes
Q2: On average, how many patients are being examined by doctors? 6.25 patients
Q3: On average, how many patients are in the ER? 77.9 patients

Little's Law example

L $\qquad\qquad\qquad$ λ $\qquad\qquad\qquad$ W

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[units] [units/time] [time]



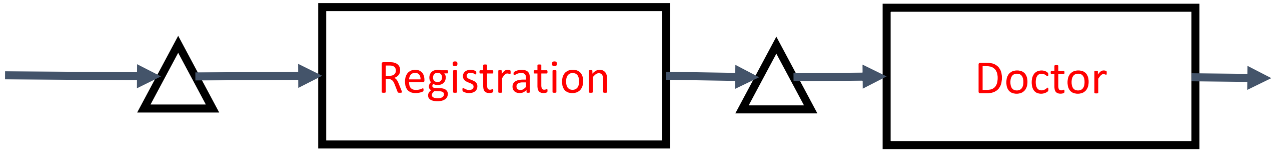
λ (people/min)	5/6	5/6	5/6	5/6
W (min)		2		7.5
L (people)	30		40	

Little's Law example

L λ W

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[units] [units/time] [time]



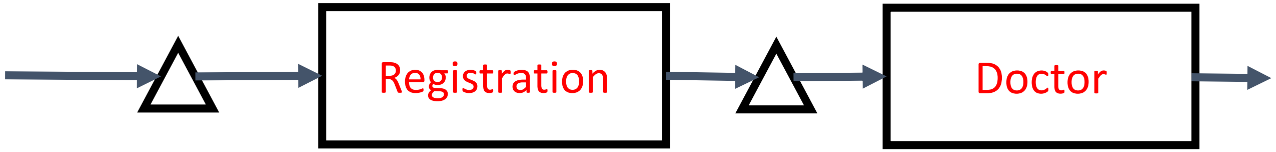
λ (people/min)	5/6	5/6	5/6	5/6
W (min)	36	2	48	7.5
L (people)	30	1.66	40	6.25

Little's Law example

L λ W

Inventory (WIP) = Avg. thrp. rate × Avg. flow time

[units] [units/time] [time]

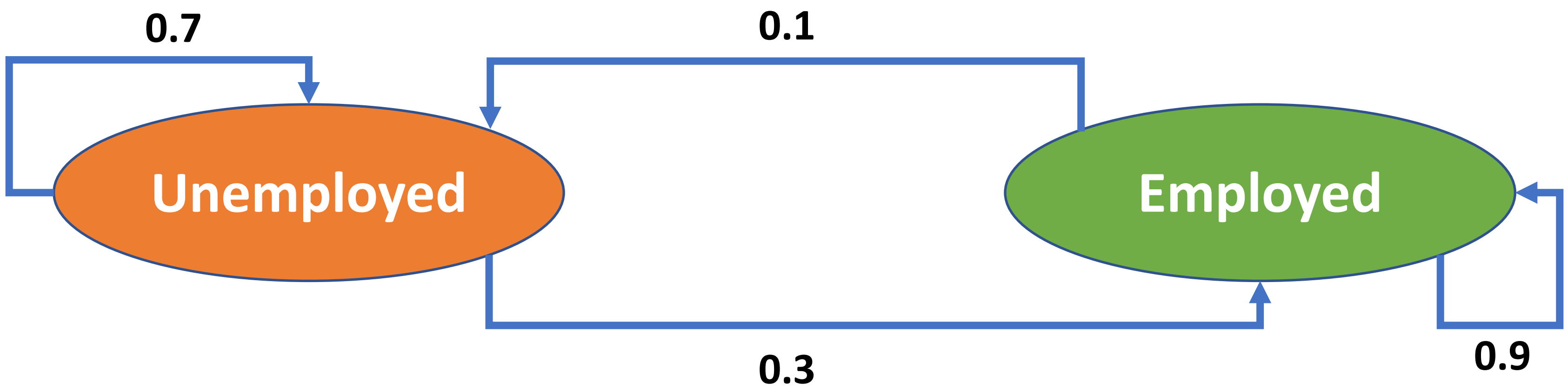


λ (people/min)	5/6	5/6	5/6	5/6	
W (min)	36	2	48	7.5	93.5
L (people)	30	1.66	40	6.25	77.9

Markov chain definition

- **Markov chains:** a modeling framework for stochastic dynamics
- The **time** is discrete: $t = 0, 1, 2, \ldots$
- We specify a discrete **state space** S
  - Examples: goals in a football game, people in a queue
- Next, we specify a family of random variables X_t with values in S
  - First, the **initial distribution** X_0
  - Next, the **transition probabilities** $p_{ij} = \mathbb{P}(X_t = j \mid X_{t-1} = i)$
  - These transition probabilities do not depend on t, only on the *values* of X_t and X_{t-1}
  - Example: probability for home team to score a goal given that the goal difference is 2:1
  - What if you want to look two periods back? Change the state space!

Simplest Markov chain example



- X_t: a person's employment status at time t
- **State space:** $\{$Employed, Unemployed $\}$
- **Transition probabilities:** $p_{EU} = 0.1, p_{UE} = 0.3$

Augmenting the state space

Suppose that now you want the following:

- If a person was unemployed for two consecutive periods, the probability of him getting a job is 0.2
- If a person was employed for two consecutive periods, the probability of him losing his job is 0.1
- Otherwise, the probability of a person being unemployed in the next period is 0.3
- How would you write down the probability space?

Markov chains: marginals

- Suppose you know the probabilities $\mathbb{P}(X_t = i) = q_i$
- What are the probabilities that $\mathbb{P}(X_{t+1} = j)$?
- Use law of total probability:

$$\mathbb{P}(X_{t+1} = j) = \sum_{i \in S} \mathbb{P}(X_{t+1} = j \mid X_t = i)\mathbb{P}(X_t = i).$$

- Or, using our previous notation:

$$\mathbb{P}(X_{t+1} = j) = \sum_{i \in S} q_i p_{ij}.$$

- In matrix notation:

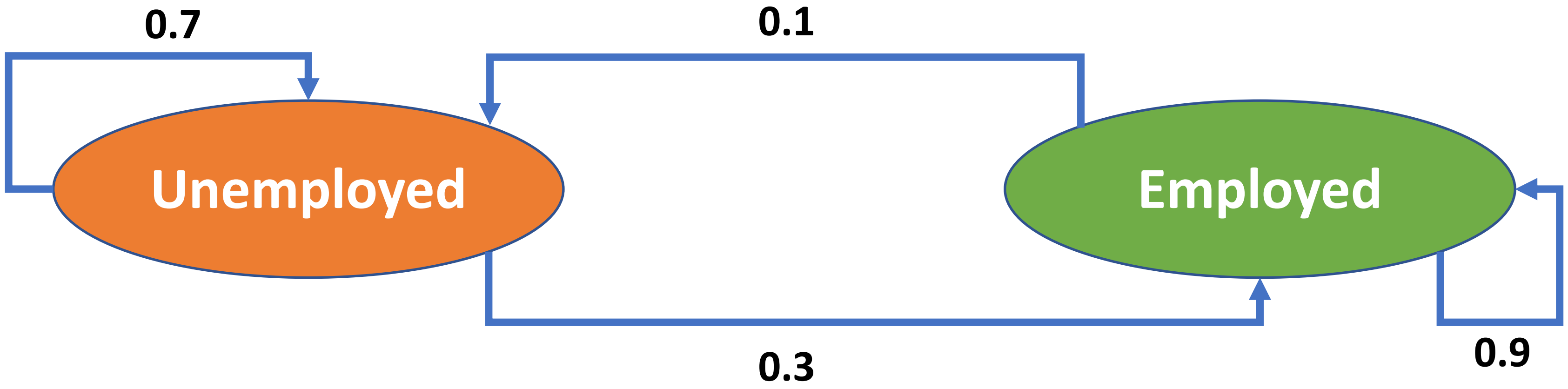
$$\mathbb{P}(X_{t+1} = i) = (\mathbf{qP})_i,$$

  where **P** is a matrix with ij-th entry equal to p_{ij}.

- And so

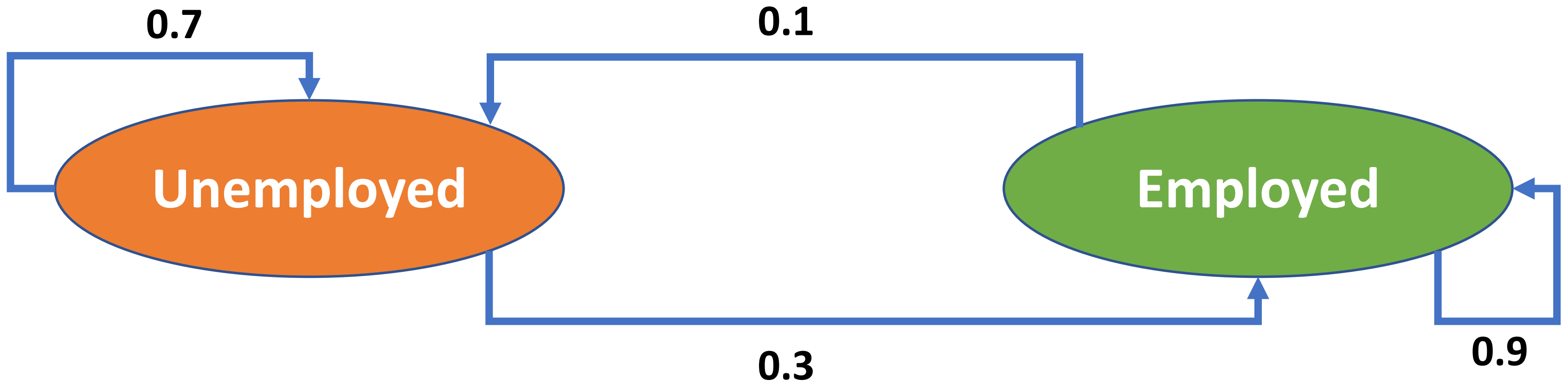
$$\mathbb{P}(X_{t+m} = i) = (\mathbf{qP}^m)_i.$$

Markov chains: marginals example



- X_t: a person's employment status at time t
- **State space:** $\{$Employed, Unemployed $\}$
- **Transition probabilities:** $p_{EU} = 0.1, p_{UE} = 0.3$
- $P(X_0 = U) = 0.6$
- What is $P(X_1 = U)$?

Markov chains: marginals example



- X_t: a person's employment status at time t
- **State space:** $\{$Employed, Unemployed $\}$
- **Transition probabilities:** $p_{EU} = 0.1, p_{UE} = 0.3$
- $P(X_0 = U) = 0.6$
- What is $P(X_1 = U)$?
- Well, it is equal to $0.6 \times 0.7 + 0.4 \times 0.1 = 0.46$.

Markov chains: stationary distribution

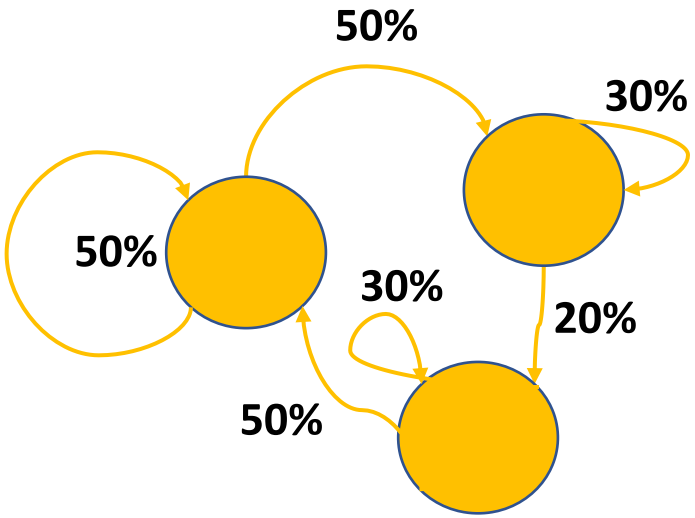
- **Stationary distribution:** a probability distribution π such that $\pi = \pi P$
- So it doesn't change under the Markov chain dynamics
- It always exists if the Markov chain is **ergodic**, meaning that it is
  - **Irreducible:** each state can be reached from every other state in finite amount of steps with positive probability.
  - **Aperiodic:** the *period* of all states in the chain equals 1.
    - The period of a state i is the GCD of all positive-probability return times from i to i
- If the chain is ergodic, it does always converge to a stationary distribution π:

$$\lim_{n \to \infty} qP^n = \pi,$$

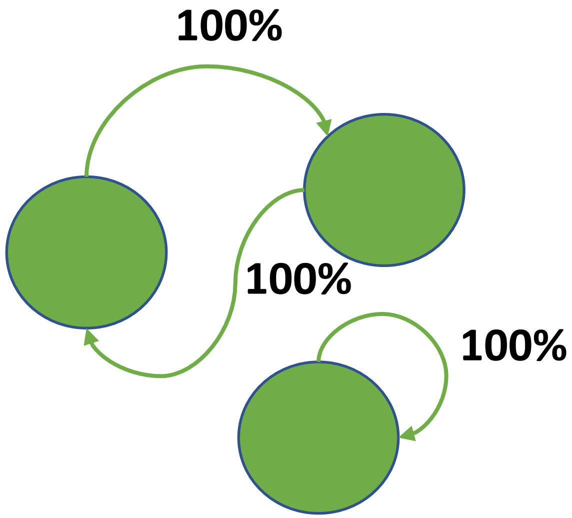
  for any q

Markov chains: stationary distribution

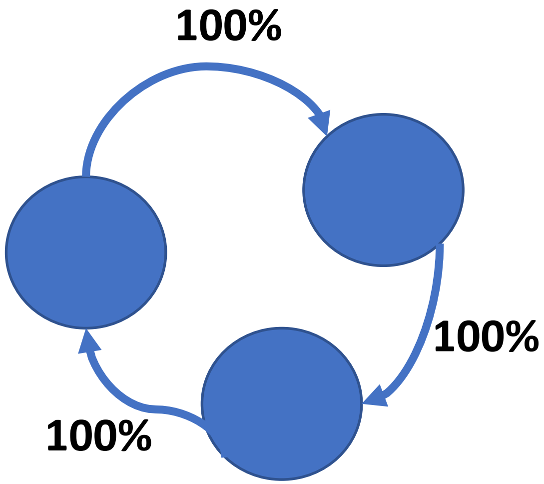
IRREDUCIBLE, APERIODIC



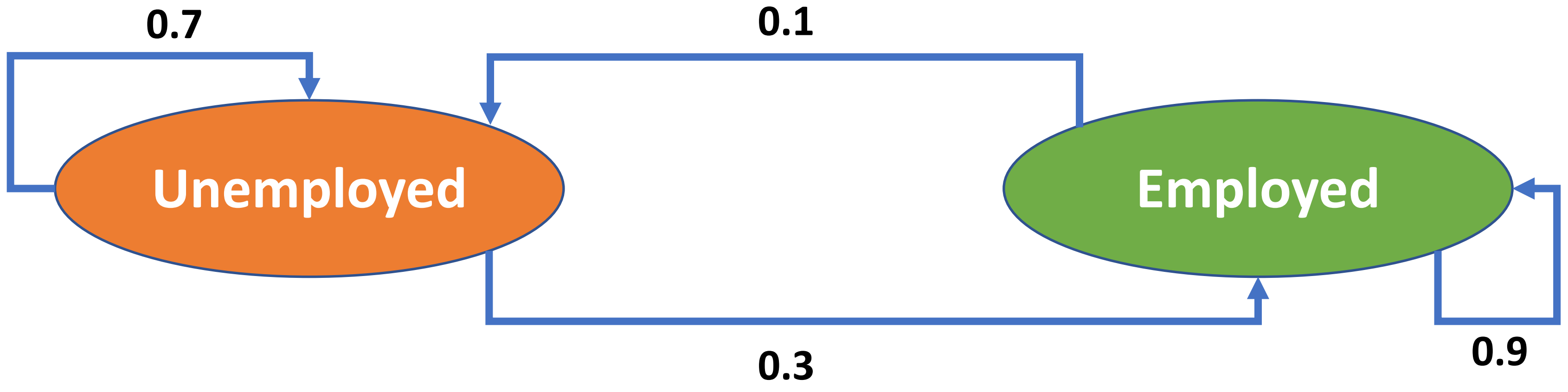
REDUCIBLE, PERIODIC



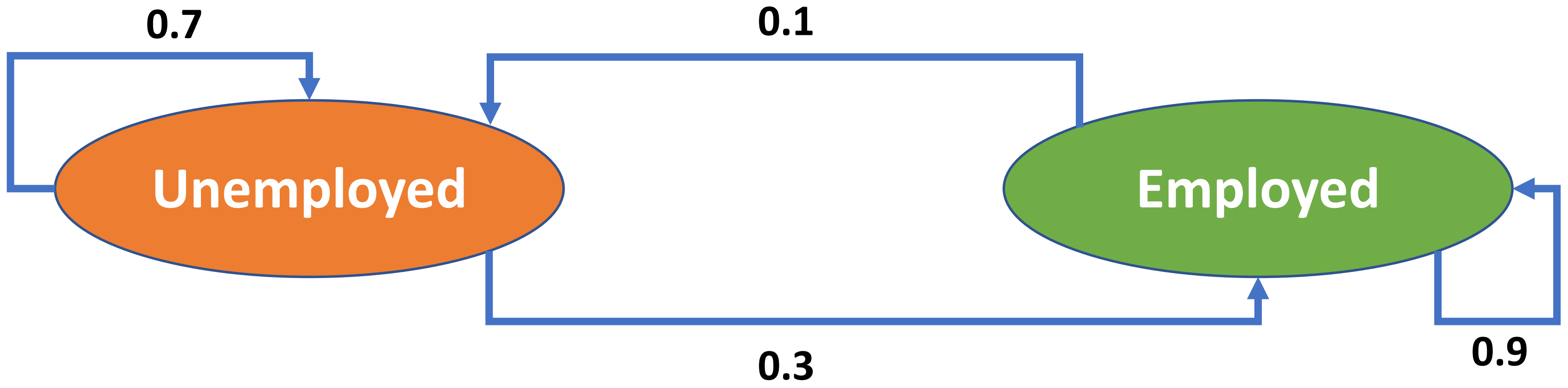
IRREDUCIBLE, PERIODIC



Markov chains: stationary distribution example



Markov chains: stationary distribution example



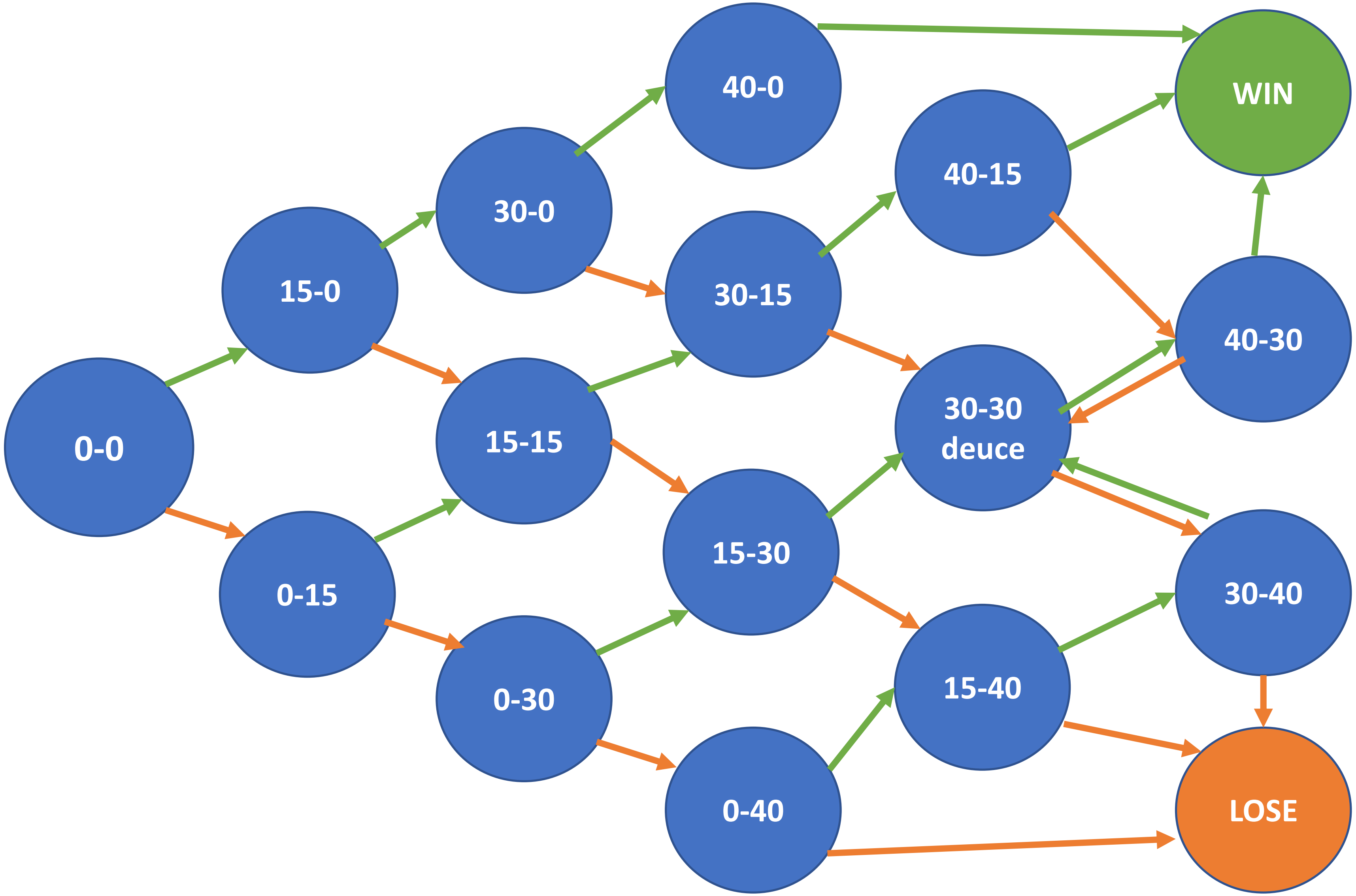
$$0.7\pi_U + 0.1\pi_E = \pi_U$$

$$0.3\pi_U + 0.9\pi_E = \pi_E$$

$$\pi_E = 3/4, \pi_U = 1/4$$

Sports analytics

A Markov chain for tennis **game.** Green arrows: home scored, orange arrows: away scored.



What can you do with this?

Specify probabilities for transitions.

Either the same for all arrows of the same color.

Or different across arrows!

How? Look at empirical frequencies.

Once you have those:

- Can predict the probability of a given outcome happening via **backward recursion**

- Or predict the eventual distribution of outcomes via **forward recursion**

Inventory management example

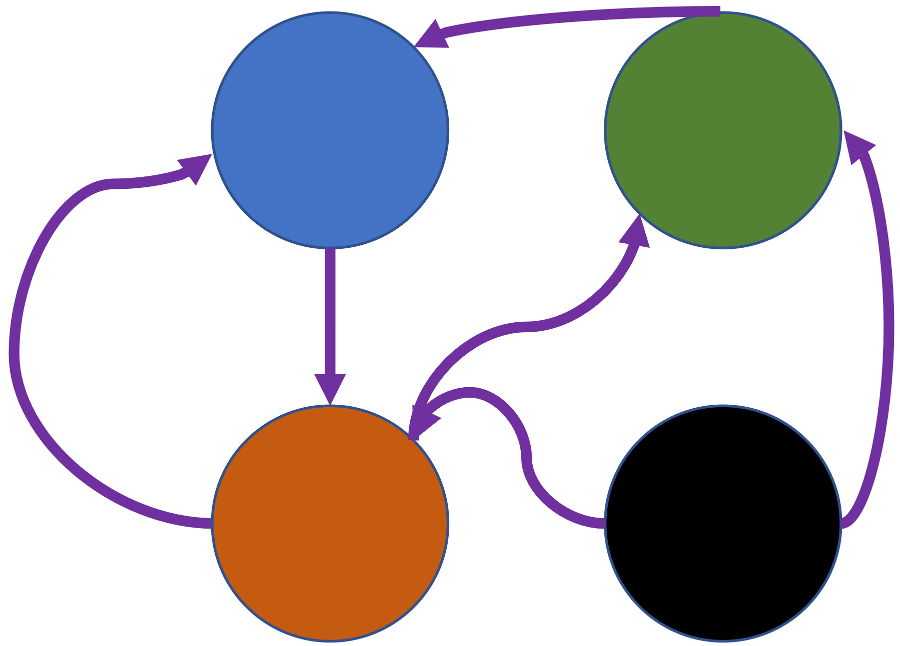
- State X_t: number of units in stock
- Capacity of the warehouse: K
- In each period:
  - A unit is delivered to the warehouse with prob. p
  - A unit is bought from the warehouse with prob. q
  - No capacity at warehouse: unit rejected
  - No units at warehouse: none sold and no backlogs.

Inventory management example

- State X_t: number of units in stock
- Capacity of the warehouse: K
- In each period:
  - A unit is delivered to the warehouse with prob. p
  - A unit is bought from the warehouse with prob. q
  - No capacity at warehouse: unit rejected
  - No units at warehouse: none sold and no backlogs.
- This is a Markov process
- **Transition rules:**
  - $0 < s < K$: $p_{ss} = pq + (1-p)(1-q)$, $p_{s,s+1} = p(1-q)$, $p_{s,s-1} = (1-p)q$
  - $s = 0$: $p_{ss} = pq + (1-p)(1-q) + (1-p)q$, $p_{s,s+1} = p(1-q)$
  - $s = K$: $p_{ss} = pq + (1-p)(1-q) + p(1-q)$, $p_{s,s-1} = (1-p)q$
- Chain is ergodic; try finding its stationary distribution at home.

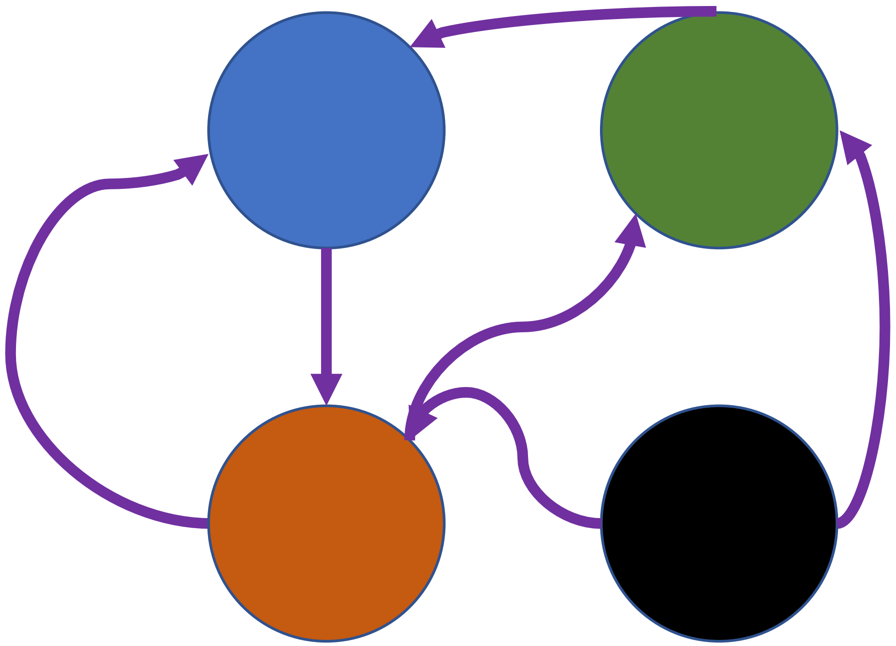
Customer preference example

Customers have favorite colors, but they change from period to period
All customers change their colors in the same way



Customer preference example

Customers have favorite colors, but they change from period to period stochastically
All customers change their colors in the same way

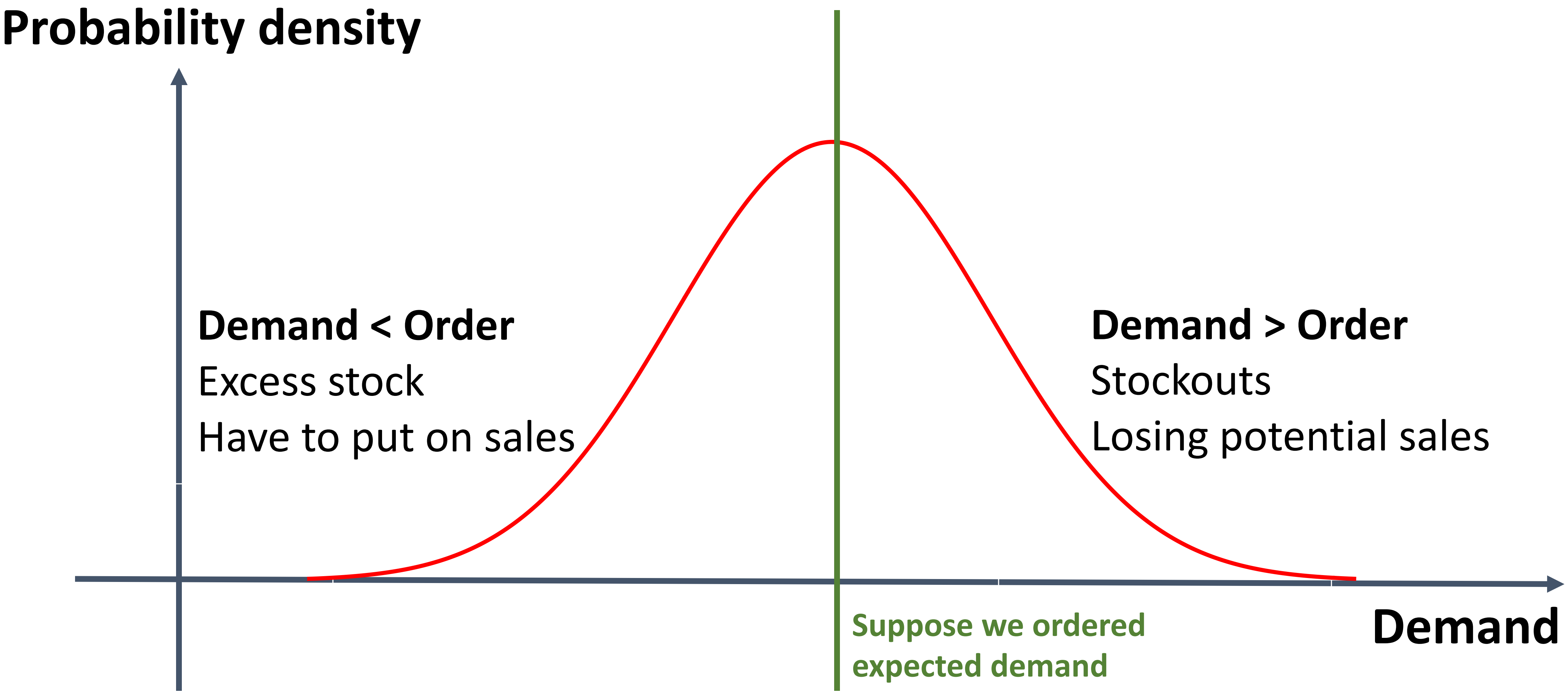


After several periods pass,
unless you observe the customer preferences
you cannot **predict exactly** what their preference is.

At best you can come up with
a **distribution** of favorite colors,
which will be close to a stationary distribution.

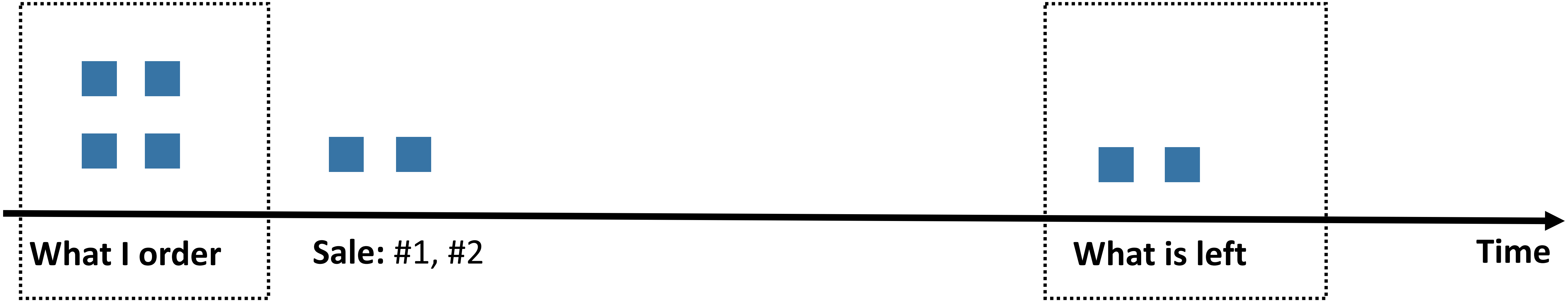
So you will face uncertainty while deciding on production quantities.

The gamble of ordering

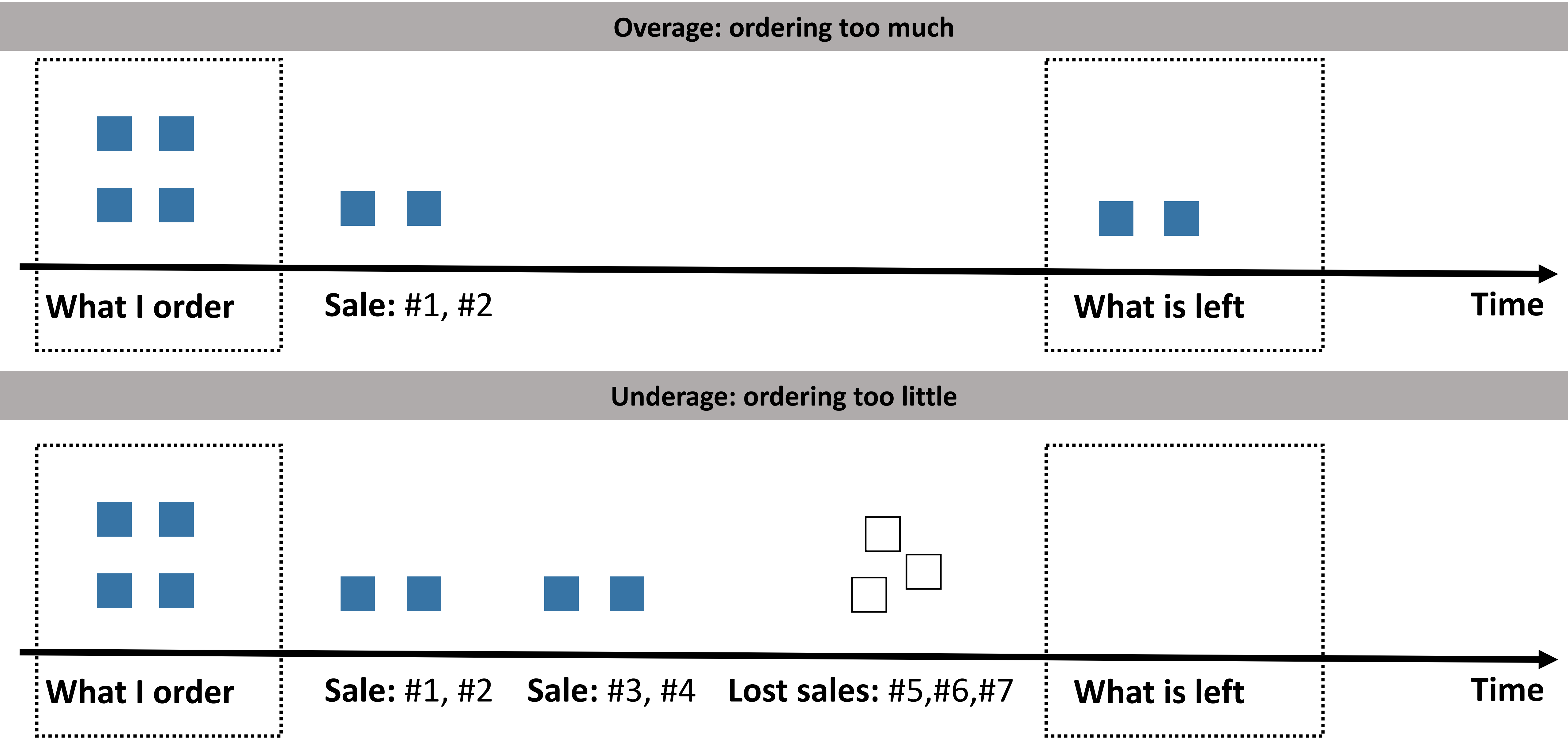


Ordering too much vs. too little

Overage: ordering too much



Ordering too much vs. too little



A simple example with three scenarios

Order quantity 1200 units

	Demand	Sales	Shortages	Leftovers
60%	600	600	0	600
30%	1200	1200	0	0
10%	2400	1200	1200	0
Expected	960	840	120	360

Newsvendor model

- Every day a newsvendor buys newspapers to sell during the day
- The newspapers lose all their value next day
- The newsvendor does not know in advance how much he will sell
- The newsvendor can:
  - purchase newspapers wholesale at price **c =** $2 a piece
  - sell current newspapers at price **p =** $5 a piece
  - salvage the old newspapers at price **s** = $1 a piece
- Demand for newspapers each day: random
  - Normal distribution
  - Mean 100, std. dev. 30
- If he picks up
  - too much: loses out on unsold ones
  - too little: loses out on unsatisfied customers
- How many newspapers should he purchase?

Newsvendor model

- Every day a newsvendor buys newspapers to sell during the day
- The newspapers lose all their value next day
- The newsvendor does not know in advance how much he will sell
- The newsvendor can:
  - purchase newspapers wholesale at price **c =** \$2 a piece
  - sell current newspapers at price **p =** \$5 a piece
  - salvage the old newspapers at price **s** = \$1 a piece
- Demand for newspapers each day: random
  - Normal distribution
  - Mean 100, std. dev. 30
- If he picks up
  - too much: loses out on unsold ones
  - too little: loses out on unsatisfied customers
- How many newspapers should he purchase?

Overage cost
Each unit in excess of daily demand
costs him **c – s** = \$1

Underage cost
Each unit of sales lost due to insufficient inventory
costs him **p – c** = \$3

Marginal analysis

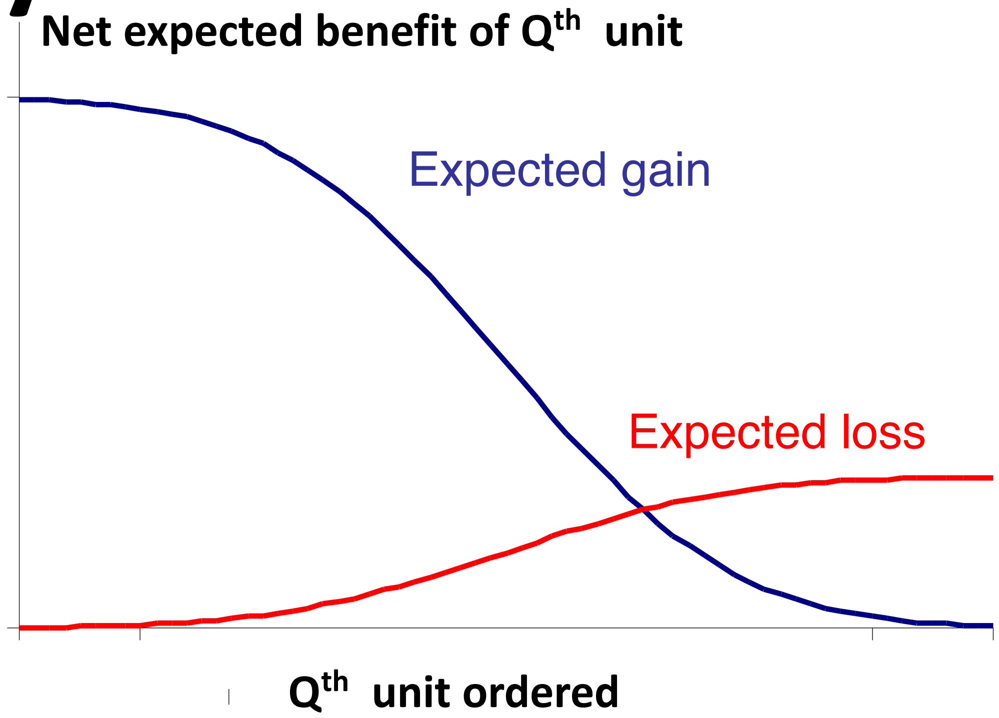
- Ordering one more unit increases the chance of overage:
  - Expected loss on Q-th unit: $c_o \times \mathbb{P}(D \leq Q)$
- But reduces the chance of underage:
  - Expected gain on Q-th unit: $c_u \times \mathbb{P}(D \geq Q)$
- To maximize expected profit: make one equal to another

$$c_o \times \mathbb{P}(D \leq Q) = c_u \times \mathbb{P}(D \geq Q)$$

- Rearranging the terms:

$$\mathbb{P}(D \leq Q) = \frac{c_u}{c_u + c_o}$$

- The expression $\frac{c_u}{c_u+c_o}$ is called the *critical ratio* (or fractile)
- To maximize profit, select Q such that the probability to satisfy all demand (or *service level* equals the critical ratio.



$$\min_Q \mathbb{E}_D \left(c_u(D-Q)^+ + c_o(Q-D)^+\right)$$

Solving our newsvendor example

1. Find the per-unit underage costs c_u and overage costs c_o.
   - In our example, $c_u = \$3$ and $c_o = \$1$.
2. Compute the critical ratio:
   - $$\frac{c_u}{c_u + c_o} = 0.75.$$
3. Now we need to compute Q^* such that

$$P(D \leq Q^*) = \frac{c_u}{c_u + c_o}.$$

But that's the same as having

$$P\left(\frac{D - \mu}{\sigma} \leq \frac{Q^* - \mu}{\sigma}\right) = \frac{c_u}{c_u + c_o}.$$

This in turn means that the optimal order quantity is equal to

$$Q^* = \mu + \sigma z^*,$$

where z^* is the $\frac{c_u}{c_u+c_o}$-th quantile of the standard Normal distribution.

- $z^* \approx 0.67$
- $Q^* = 100 + 30 \times z^* = 120$

Performance analysis

Unit of Demand

Is Satisfied → **Sales**

Is **NOT** Satisfied → **Shortages**

Expected sales = exp. demand – exp. shortages

Unit of Inventory Ordered

Is Bought → **Sales**

Is **NOT** Bought → **Leftover**

Expected leftovers = order size – exp. sales

Expected shortages computation

If we know the expected shortages, we're able to compute everything else. For normally distributed demand, we can compute the expected shortages explicitly. Writing down the expression formally, we get that:

$$\text{Expected shortages} = \mathbb{E}_D \max((D - Q^*), 0).$$

But we know that D = $\mu + \sigma$Z, where Z is a standard normal random variable; also, we know that Q* = $\mu + \sigma$z* where z* is a quantile of the standard normal distribution corresponding to critical ratio. Plugging this in, the expression simplifies further:

$$\text{Expected shortages} = \sigma \mathbb{E}_Z \max((Z - z^*), 0).$$

Integrating the expectation, we get:

Std. norm. PDF

$$\mathbb{E}_Z \max((Z - z^*), 0) = \phi(z^*) - z^*(1 - \Phi(z^*)),$$

Std. norm. CDF

This function is typically denoted L(z*) and is called the standard loss function. Tables for it exist. So we derived that the expected shortages are given by σL(z*) where L($\cdot$) is the standard loss function.

Performance analysis: expressions

General expressions for normally distributed demand

1. Expected shortages: σL(z) (where L(z) = φ(z) $-$ z(1 $-$ Φ(z)))
2. Expected sales: $\mu - \sigma$L(z)
3. Expected leftover inventory: $Q^* -$ Expected sales
4. Expected profits: $(r-c)\times\mu-c_u\times$Expected shortages$-c_o\times$Expected leftover inventory

Our old example

1. Expected shortages: $\sigma L(z^*) = 30 \times 0.15 = 4.5$ units.
2. Expected sales: $100 - 4.5 = 95.5$ units.
3. Expected leftover inventory: $120 - 95.5 = 24.5$ units.
4. Expected profits: $3 \times 100 - 3 \times 4.5 - 24.5 = 262$ (\$).

Helpful Python functions

```
# Examples of python functions
import scipy.stats as ss
mu, sigma, Q = 100, 30, 120
c_u, c_o = 3, 1

# norm.cdf: Normal CDF. P(D < Q) if D is N(mu, sigma).
Phi = ss.norm.cdf(Q, mu, sigma)
Phi 0.7475074624530771
# norm.pdf: Normal PDF. Derivative of normal CDF.
phi = ss.norm.pdf(Q, mu, sigma)
phi 0.010648266850745075
# norm.ppf: Normal inverse-CDF. Find z-score given critical fractile
z = ss.norm.ppf(c_u/(c_u + c_o))
z 0.6744897501960817
# Normal loss function
L = lambda z: ss.norm.pdf(z) - z * (1 - ss.norm.cdf(z))
# Using it:
shortage = sigma * L(z)
shortage 4.474624054052596 6
```

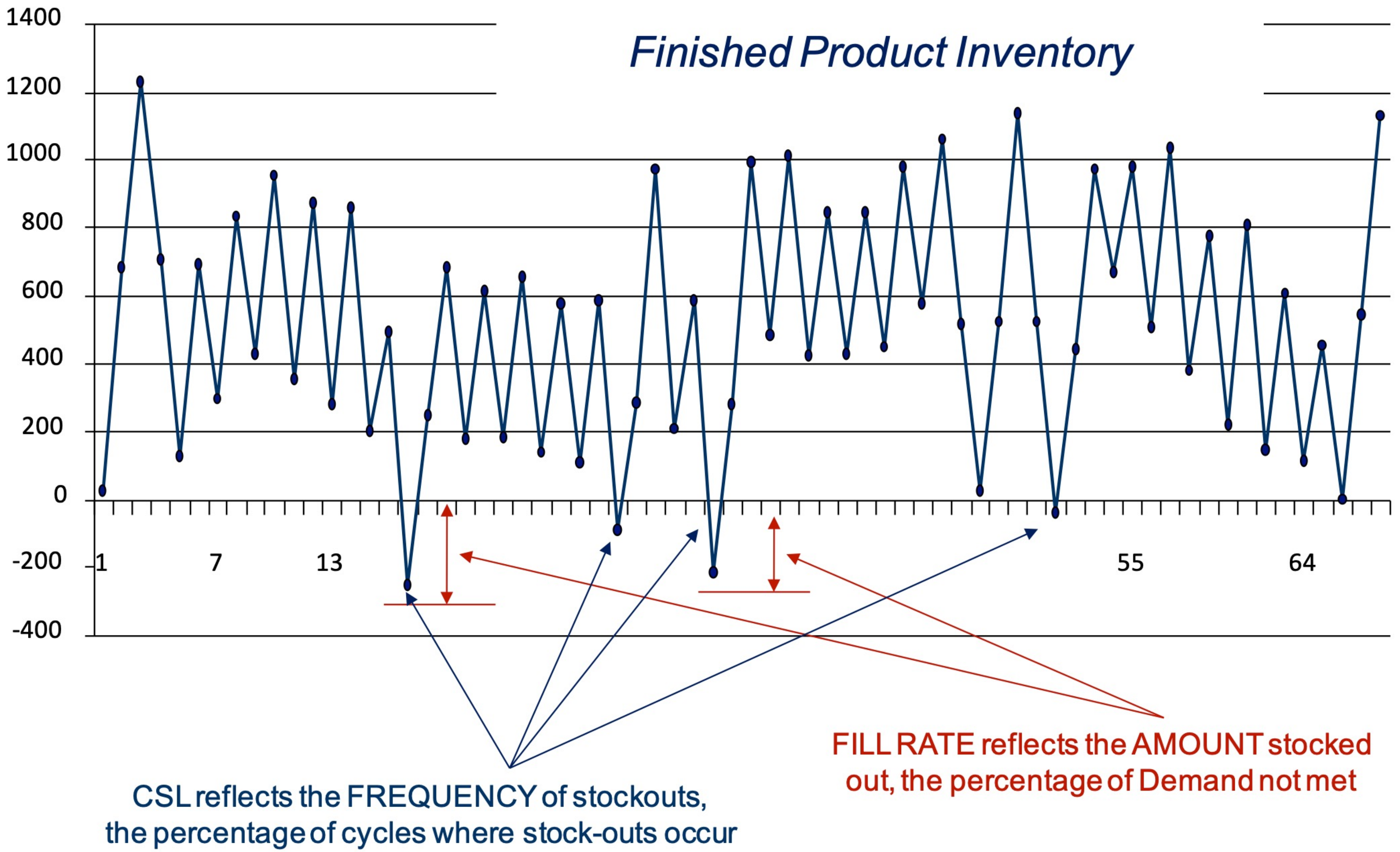
Two metrics of availability

- *Cycle service level*: the probability of satisfying the demand. Equal to critical ratio if the order quantity is optimal. Represents the **frequency of stock-outs**.

- *Expected fill rate*: percentage of demand filled immediately from inventory. Represents the **magnitude of stock-outs**.

$$\text{Fill rate} = \frac{\text{Exp. demand} - \text{Exp. shortages}}{\text{Exp. demand}} = \frac{\text{Exp. sales}}{\text{Exp. demand}}$$

Fill rate vs. service level



Fill rate vs. service level

observation	inventory	demand	Demand Satisfied?	Shortage
1	45	13	1	0
2	45	139	0	94
3	45	40	1	0
4	45	16	1	0
5	45	28	1	0
6	45	98	0	53
7	45	18	1	0
8	45	7	1	0
9	45	22	1	0
10	45	10	1	0
		Total:	**Total:**	**Total:**
		391	8	147
		Average:	**Average:**	**Average:**
		39.1	0.8	14.7
		Service Level = 0.8		
		Fill Rate = (39.1 - 14.7)/39.1 = 0.62		

Newsvendor model as an LP

- Let ω be a scenario, and $\mathcal{O}$ be the set of all scenarios
- Let $p(\omega)$ be the probability of scenario ω and $D(\omega)$ the demand value
- Let $z(\omega)$ be the decision variable: the amount sold in scenario ω
- Let Q be the amount purchased in advance (one cannot sell more than that)
- Let r be the retail selling price, and c the purchasing cost

Scenario	Probability	Demand
1	0.7	100
2	0.1	40
3	0.2	190

$$
\begin{aligned}
\max_{z(\omega),Q} \quad & r \sum_{\omega \in \mathcal{O}} p(\omega) z(\omega) - cQ \\
\text{s.t.} \quad & z(\omega) \leq D(\omega) \ \text{ for all } \omega \in \mathcal{O} \\
& z(\omega) \leq Q \ \text{ for all } \omega \in \mathcal{O} \\
& z(\omega) \geq 0 \text{ for all } \omega \in \mathcal{O} \\
& Q \geq 0
\end{aligned}
$$

Newsvendor model as an LP

Scenario	Probability	Demand
1	0.7	100
2	0.1	40
3	0.2	190

- Let ω be a scenario, and $\mathcal{O}$ be the set of all scenarios
- Let $p(\omega)$ be the probability of scenario ω and $D(\omega)$ the demand value
- Let $z(\omega)$ be the decision variable: the amount sold in scenario ω
- Let Q be the amount purchased in advance (one cannot sell more than that)
- Let r be the retail selling price, and c the purchasing cost

$$
\begin{aligned}
\max_{z(\omega),Q} \quad & r \sum_{\omega \in \mathcal{O}} p(\omega) z(\omega) - cQ \\
\text{s.t.} \quad & z(\omega) \leq D(\omega) \ \text{ for all } \omega \in \mathcal{O} \\
& z(\omega) \leq Q \ \text{ for all } \omega \in \mathcal{O} \\
& z(\omega) \geq 0 \text{ for all } \omega \in \mathcal{O} \\
& Q \geq 0
\end{aligned}
$$

Newsvendor model as an LP

- Let ω be a scenario, and $\mathcal{O}$ be the set of all scenarios
- Let $p(\omega)$ be the probability of scenario ω and $D(\omega)$ the demand value
- Let $z(\omega)$ be the decision variable: the amount sold in scenario ω
- Let Q be the amount purchased in advance (one cannot sell more than that)
- Let r be the retail selling price, and c the purchasing cost

Scenario	Probability	Demand
1	0.7	100
2	0.1	40
3	0.2	190

$$
\begin{aligned}
\max_{z(\omega),Q} \quad & r \sum_{\omega \in \mathcal{O}} p(\omega) z(\omega) - cQ \\
\text{s.t.} \quad & z(\omega) \leq D(\omega) \ \text{ for all } \omega \in \mathcal{O} \\
& z(\omega) \leq Q \ \text{ for all } \omega \in \mathcal{O} \\
& z(\omega) \geq 0 \text{ for all } \omega \in \mathcal{O} \\
& Q \geq 0
\end{aligned}
$$

Two-stage stochastic programming problems follow this pattern:

1. Stage 1: decision before the uncertainty is resolved (e.g. order quantity)
2. Stage 2: decision after the uncertainty is resolved (e. g. sales quantity)

Up next

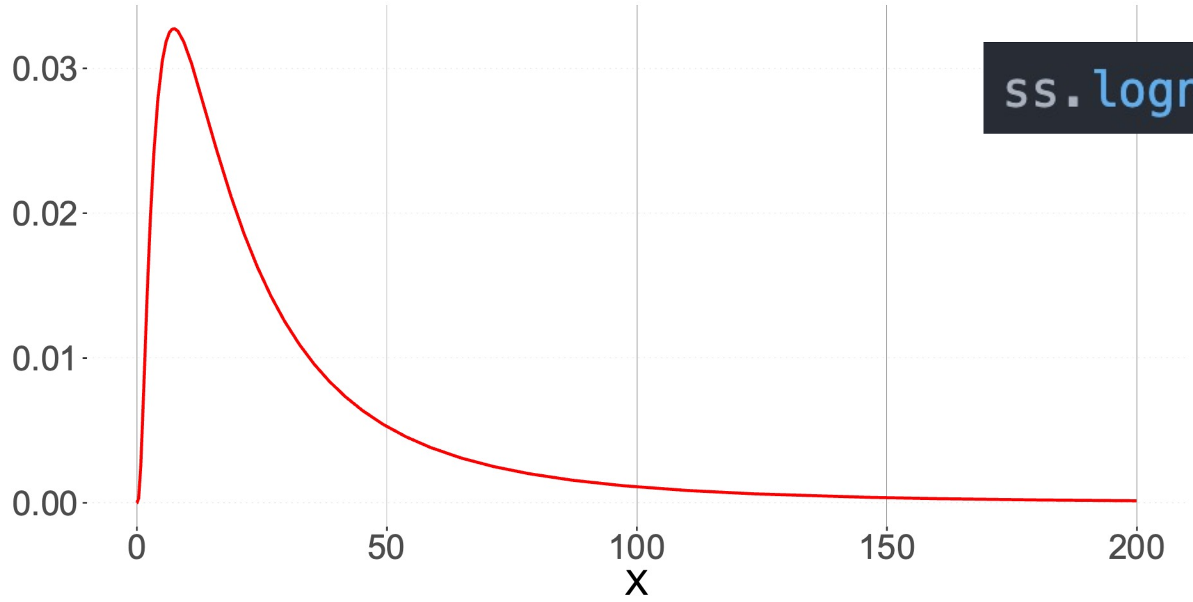
- Assignment 2 due on April 18th 23:59:59
- Seminar: how to work with Markov chains?
  - Simulating the chains
  - Understanding stationary distributions
- We will consider more complex stochastic programming problems
  - Multiple products
  - Learning demand over time
  - Pooling
  - How did Zara, Benetton, Dell, etc. use these ideas?

Business analytics I
Operations Analytics
Class 6
Applications of Stochastic Programming.

Marat Salikhov

April 18th , 2022

Lognormal distribution




```
ss.lognorm(s = sigma_s, scale = np.exp(mu_s)).rvs()
```

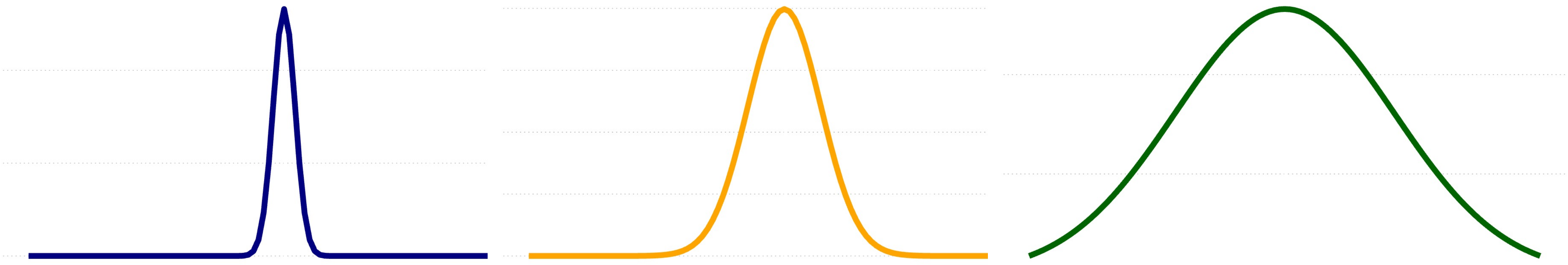
σ $\qquad$ $\exp(\mu)$

$$\log X \sim N(\mu, \sigma)$$

- μ and σ are not the mean and variance of the lognormal!
- $\mathbb{E}X = \exp\left(\mu + \frac{\sigma^2}{2}\right)$
- $\mathbb{V}X = \exp\left(2\mu + \sigma^2\right)\left[\exp(\sigma^2) - 1\right]$
- In HA 2, you need to solve for μ and σ from these expressions

Newsvendor: best demand distribution?

Which one is better?



How can demand uncertainty be reduced?

- Be fast: **reduce lead times**
- Learn demand over time: **make-to-order** and **reactive capacity**
- Combine multiple products: **risk pooling**
- Do both: **delayed differentiation**
- **All these choices are usually costly!**


ZARA
ZARA

Zara's success: why?

Zara, 25, makes debut to match M&S, 117

Spanish fashion chain Zara made a sparkling debut on the Madrid stock exchange yesterday, making founder Amancio Ortega and his family £1.5bn richer.

Shares in Zara's parent company, Inditex, jumped 22% to €11.2bn (£6.7bn), giving the 25-year-old chain a market value almost as great as the 117-year-old Marks & Spencer.

The reception confirmed expectations for the float of a company whose innovative production process and succesful stores are marking out the future for the sector in Europe.

Be fast, like Zara

Fashion product cycle

→ Design → Sourcing → Mfg. → Distribution → Sales → Salvage →

Be fast, like Zara

Fashion product cycle

→ Design → Sourcing → Mfg. → Distribution → Sales → Salvage →

MARKS & SPENCER

Marks and Spencer in 1990s

- Styles defined 1 year in advance
- Prices and quantities are set in advance
- Most styles are traditional
- Production is done overseas
- Many distributed warehouses
- Seven weeks of inventory on hand

Whole cycle length: 16 months
Avg. annual sales growth: 4.5%

Be fast, like Zara

Fashion product cycle

→ Design → Sourcing → Mfg. → Distribution → Sales → Salvage →

MARKS & SPENCER

ZARA

Marks and Spencer in 1990s

- Styles defined 1 year in advance
- Prices and quantities are set in advance
- Most styles are traditional
- Production is done overseas
- Many distributed warehouses
- Seven weeks of inventory on hand

Whole cycle length: 16 months
Avg. annual sales growth: 4.5%

Zara in 1990s

- Styles are defined flexibly
- Prices and quantities are adjusted on the fly
- Last minute changes to styles
- All production is close to markets
- One centralized warehouse
- Few days of inventory

Whole cycle took 3-4 weeks
Avg. annual sales growth: 20%

Be fast, like Zara

Fashion product cycle

Design → Sourcing → Mfg. → Distribution → Sales → Salvage

MARKS & SPENCER

ZARA

Marks and Spencer in 1990s

- Styles defined 1 year in advance
- Prices and quantities are set in advance
- Most styles are traditional
- Production is done overseas
- Many distributed warehouses
- Seven weeks of inventory on hand

Whole cycle length: 16 months
Avg. annual sales growth: 4.5%

Zara in 1990s

- Styles are defined flexibly
- Prices and quantities are adjusted on the fly
- Last minute changes to styles
- All production is close to markets
- One centralized warehouse
- Few days of inventory

Whole cycle took 3-4 weeks
Avg. annual sales growth: 20%

Inditex's caution could be wise. Only two years ago it missed some of the year's main fashion trends. In their assessment of what went wrong in 2003, analysts at CSFB, an investment bank, identified issues of complexity and control as being among the causes—worrying for a company that likes to keep things simple. Even so, José Luis Nueno of IESE, a business school in Barcelona, believes the firm will grow successfully. Consumers have become more demanding and more arbitrary, so fast fashion is better suited to these changes, he argues. Inditex has proven that cheap and fast fashion can also be trendy and well presented. Zara's stores have won awards for their decoration and their shop windows.

To smooth its growth, Inditex has opened a new distribution centre in Zaragoza. It has also begun to obtain some of its basic garments from low-cost countries, although the bulk of its production remains in Europe. China, for instance, accounts for just 12.5% of its production, less than that of rivals. Yet the further Inditex moves away from home, the trickier it will be to cater to instant-fashion whims. When Madonna gave a series of concerts in Spain, teenage girls were able to sport at her last performance the outfit she wore for her first concert, thanks to Zara. Although Mr Castellano insists that Inditex can be as nimble in other countries as it is at home, that could prove difficult. And if it stumbles, as Mr Castellano must know, the fickle world of fashion will be merciless.

Make-to-stock vs. make-to-order

Make-to-stock

- Company produces in advance
- Customer does not wait
- Model with newsvendor
- Examples
  - Buffet tables
  - Ready-to-wear clothes
  - Vending machines
- Good when customers can't wait

Make-to-stock vs. make-to-order

Make-to-stock

- Company produces in advance
- Customer does not wait
- Model with newsvendor
- Examples
  - Buffet tables
  - Ready-to-wear clothes
  - Vending machines
- Good when customers can't wait

Make-to-order

- Customers orders first
- Then wait for it to arrive
- Model with queueing theory
- Examples
  - Customized laptops
  - Tailored suits
  - Airlines buying planes
- Good when customers want variety

Make-to-stock vs. make-to-order example

A limited edition notebook will be sold for 100 days only, and it will come in two colors: Red and Black. For each color, the standard deviation of demand is 20 units, the unit selling price is $50, and the unit purchase cost is $20. All notebooks that are unsold after the season ends are destroyed (essentially for free).

Make-to-stock vs. make-to-order example

A limited edition notebook will be sold for 100 days only, and it will come in two colors: Red and Black. For each color, the standard deviation of demand is 20 units, the unit selling price is $50, and the unit purchase cost is $20. All notebooks that are unsold after the season ends are destroyed (essentially for free).

Make-to-stock

- Expected value of demand for each color: 100 units
- Optimal order quantity for each color: 105 units
- Expected shortages for each color: 5.7 units
- Total expected profit: **$5,226**

Make-to-stock vs. make-to-order example

A limited edition notebook will be sold for 100 days only, and it will come in two colors: Red and Black. For each color, the standard deviation of demand is 20 units, the unit selling price is $50, and the unit purchase cost is $20. All notebooks that are unsold after the season ends are destroyed (essentially for free).

Make-to-stock

- Expected value of demand for each color: 100 units
- Optimal order quantity for each color: 105 units
- Expected shortages for each color: 5.7 units
- Total expected profit: **$5,226**

Make-to-order

- Customers have to wait for exactly 3 weeks, and they don't like it
- Because of that, the expected value of demand for each color is now 90 units
- But there are no leftovers or shortages!
- So we will sell to all customers
- Total expected profit: $90*2*(50-20) = **$5,400**

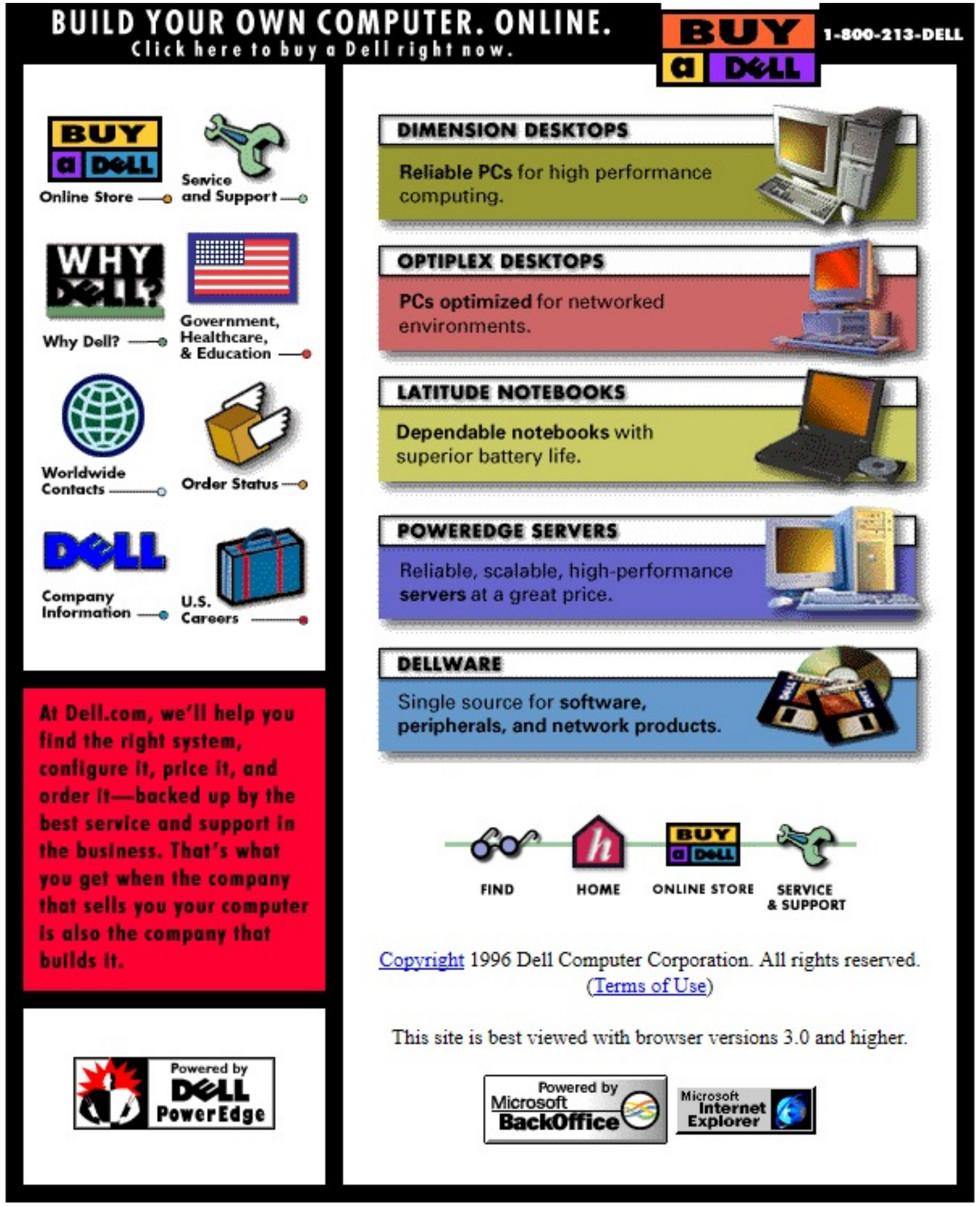
Make-to-order: Dell computers

- Computers in the 1990s:
  - Rapidly becoming obsolete
  - So huge overage costs
  - Plenty of customization options
  - How do you even forecast them?!
- Dell's idea: **don't**.
  - Instead, sell via website
  - Allow the customers to customize
  - Ignore retail

Make-to-order: Dell computers

- Computers in the 1990s:
  - Rapidly becoming obsolete
  - So huge overage costs
  - Plenty of customization options
  - How do you even forecast them?!

- Dell's idea: **don't**.
  - Instead, sell via website
  - Allow the customers to customize
  - Ignore retail



Make-to-order: Dell in 1994

THE NEW YORK TIMES, TUESDAY, JULY 12, 1994

Dell Computer to Abandon Retail Store Sales

By MATTHEW L. WALD

Seeking to play to its strengths, the Dell Computer Corporation said yesterday that it would stop selling computers in stores and concentrate instead on direct sales to consumers.

The move is expected to have little overall impact on Dell. The company has already pared its retail business to roughly 2 percent of its annual sales of $3.5 billion, and the operation is not profitable, Michael S. Dell, chairman and chief executive, said in a telephone interview.

And Dell computers will still be readily available, he said, adding, "There are 180 million telephones in the United States, and anyone can reach us."

The main reason for eliminating retail sales is to reduce costs and pricing risk, Mr. Dell said. Conventional retailing is too slow for a product like computers, which have a short shelf life, he said.

Intel Lowers Chip Prices

"Intel just announced that prices of chips are going down," he said. "If you've got chips sitting on the shelf from 30 days ago, or if the manufacturer has 70 days of inventory, you can't respond as quickly to changes," he said. With no inventory in warehouses, Dell can adjust prices faster, he said.

How the Hardware Moves

Personal computer sales in United States in the first quarter of 1994.

Manufacturer	Units sold	Share of total sold directly to consumers (nonretail)
Compaq	475,000	Less than 5 percent
Packard Bell	430,000	None
Apple	408,000	None
IBM	360,000	About 5 percent
Gateway 2000	219,000	Almost all
Dell	168,000	Nearly 90 percent
AST	141,000	None

source: IDC

The New York Times

Dell ranks sixth in personal computer sales in the United States, behind Gateway 2000, which also specializes in direct sales. A year ago it was ahead of Gateway, according to Richard Zwetchkenbaum, of the International Data Group, a consulting firm in Framingham, Mass. The leaders, Compaq Computer, Packard Bell, Apple Computer and I.B.M., all sell their machines primarily in stores.

Mr. Zwetchkenbaum and other analysts said that at least in the short term Dell's move was probably a good one, but that the company would forsake some brand identity, and thus, some future sales.

"In some respects it is a good solution for Dell," he said. "They were out of their league, in a sense," he said, adding that direct-channel sales are Dell's strong point.

But in the long term, he said, not having retail sales means not having a strong brand image, and closing off some potential buyers. About 35 percent of households in this country have a personal computer, Mr. Zwetchkenbaum said, but the others, "the great unwashed," will not buy their first machine by calling an "800" number, he said.

"The novice and the first-time buyer generally are going to want to touch and feel," he said. "They want the in-store experience. They want to hear the multimedia, with different speakers, see the different sized monitors and graphics resolutions, and talk to somebody interactively, and bring the kids along with them on the second trip."

Fred Gardner, the hardware editor of Computer Reseller News, a weekly in Manhasset, L.I., said: "They'll be getting more experienced customers, but maybe that's what they want. It's a lot less trouble for them."

Wall Street seemed to favor the move. Dell's shares gained 81.25 cents yesterday to close at $28.875 in Nasdaq trading.

Dell and other direct-sales companies offer what Dell calls "mass customization," which means letting the buyer mix-and-match components, like the central processor, the size of the storage disk, the amount of work-

Make-to-order: Dell in 1995

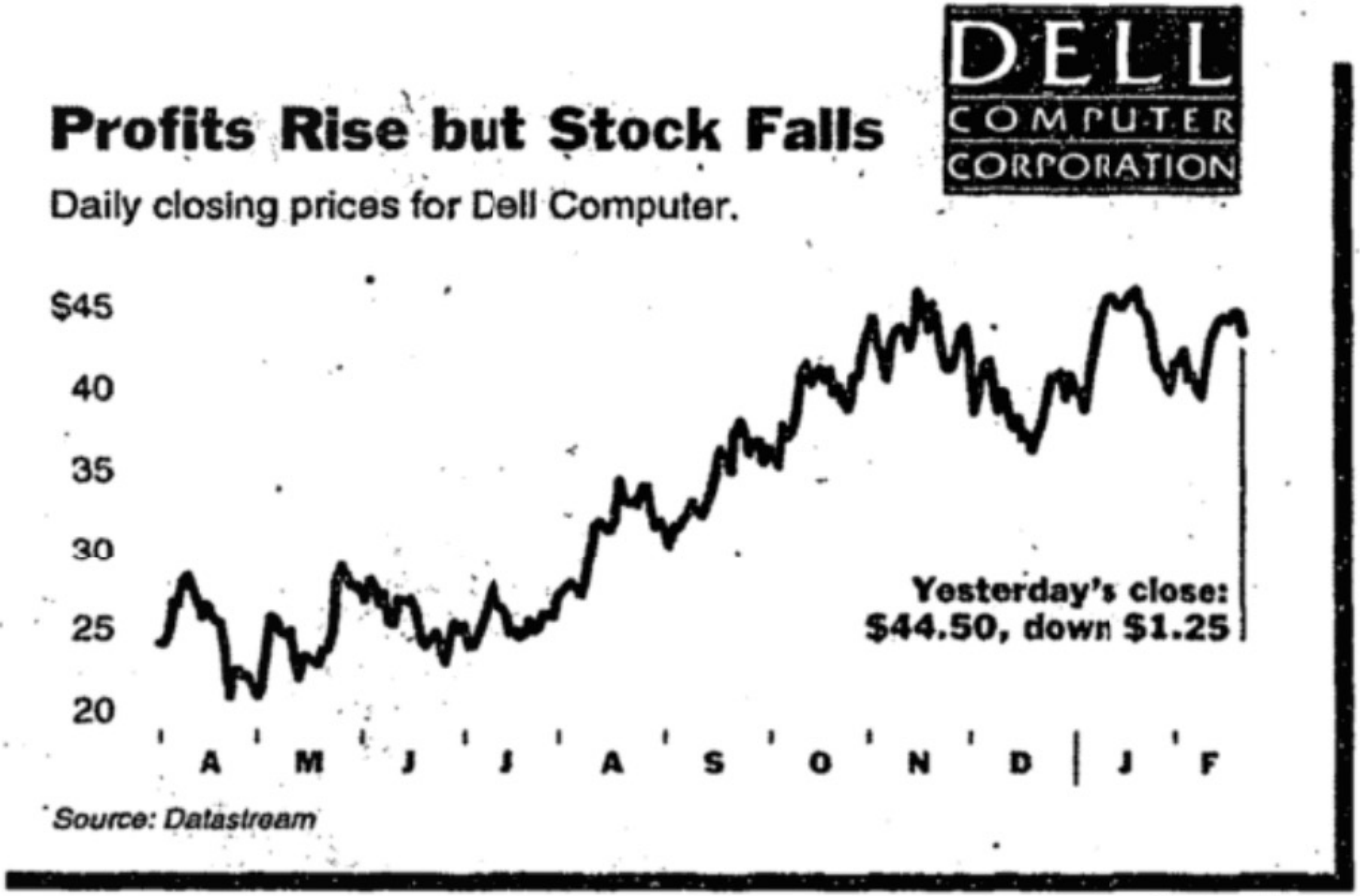
For Dell, a Tripling of Earnings

By STEVE LOHR

The Dell Computer Corporation reported yesterday that its profits tripled for the fourth quarter, helped by the revival of its notebook computer business.

The strong performance by Dell also underlines the success of its decision last spring to abandon an earlier effort to sell its personal computers through big retailers, like Wal-Mart, and to return to its roots as a mail-order PC marketer. Today, Dell ships its machines directly from its factories to buyers instead of through retail middlemen.

"People have predicted that the direct channel would shrink for years, but we see great opportunity in our business model," said Michael S. Dell, the company's 29-year-old founder and chairman.



The New York Times

Make-to-order: Dell in 2000 and 2002

Dell Computer Beats Compaq In U.S. Sales

SAN JOSE, Calif., Jan. 23 (AP) — The Dell Computer Corporation surpassed the Compaq Computer Corporation in annual United States sales for the first time last year, as personal computer sales for I.B.M. slipped, according to industry data that will be released Monday by two research firms.

But both firms report that Compaq still holds the lead in global sales.

And for the industry over all, despite concerns about the Year 2000 computing problem, PC sales surged 22 percent worldwide last year.

Dell, which is based in Round Rock, Tex., sold 7.02 million PC's for the year, taking a 16 percent share of the United States market — up from 12.7 percent in 1998, according to the research firm Dataquest, a unit of Gartner Group.

Compaq, based in Houston, sold 6.86 million computers, giving it a 15.7 percent market share, Dataquest said. Compaq's share a year earlier was 16.1 percent.

c|net

REVIEWS NEWS TECH MONEY WELLNESS HOME CARS DEALS

Dell beats Compaq for No. 1 ranking

The company surpasses Compaq in the first quarter to become the world's largest PC maker--the first time the rankings have shifted in about seven years.

Michael Kanellos Jan. 2, 2002 4:43 p.m. PT

0

Dell Computer surpassed Compaq Computer in the first quarter to become the world's largest PC maker, the first time the rankings have shifted in about seven years.

Dell achieved 12.8 percent market share worldwide in the first quarter, surpassing Compaq for the first time on a worldwide basis, according to figures released from research firm Gartner. Compaq ended the quarter in second place with a 12.1 percent share of the world market.

Make-to-stock: Dell goes back to retail

The New York Times

Dell's Founder Is Rethinking Direct Sales

By Damon Darlin

April 28, 2007

Michael S. Dell, the chairman and chief executive of Dell, who built his business by selling direct to his customers, is now thinking about changing the way the company markets its computers.

"The direct model has been a revolution, but it is not a religion," Mr. Dell wrote in a memorandum sent on Wednesday to 80,000 Dell employees.

It is the first time that Mr. Dell or any other senior executive has publicly conceded that the business model that was crucial to the company's success could — and should — be altered. Until now, the company responded with an adamant no when Wall Street analysts or customers asked whether the company would consider other

The New York Times

Dell seeking retail appeal

By Victoria Shannon

June 13, 2007

PARIS — After seeing his pioneering company lose its way over the past year or so, Michael Dell is back in the captain's chair and charting a course away from his original, single-minded mission to sell personal computers direct.

On a visit to Europe last week, Michael Dell promised that a year from now, you will have "many different ways to buy our products," not just by using the Dell site on the Internet that you are used to.

One way is to go to your favorite consumer electronics or computer store, where you will soon find Dell machines on the shelves. Michael Dell started this approach last month - after he took back his job as chief executive officer - with Wal-Mart in the United States. But he said the company would soon strike similar deals with major retailers in Europe, where Dell records more than $15 billion in annual sales.

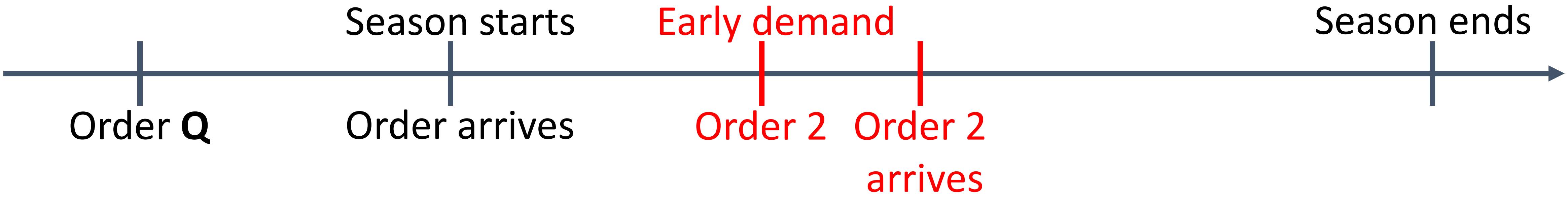
Reactive capacity

Basic newsvendor model

Reactive capacity

Basic newsvendor model

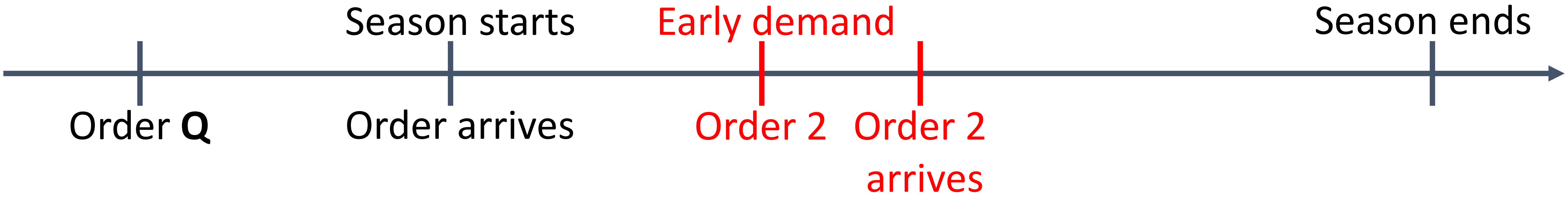
Reactive capacity



Reactive capacity

Basic newsvendor model

Reactive capacity



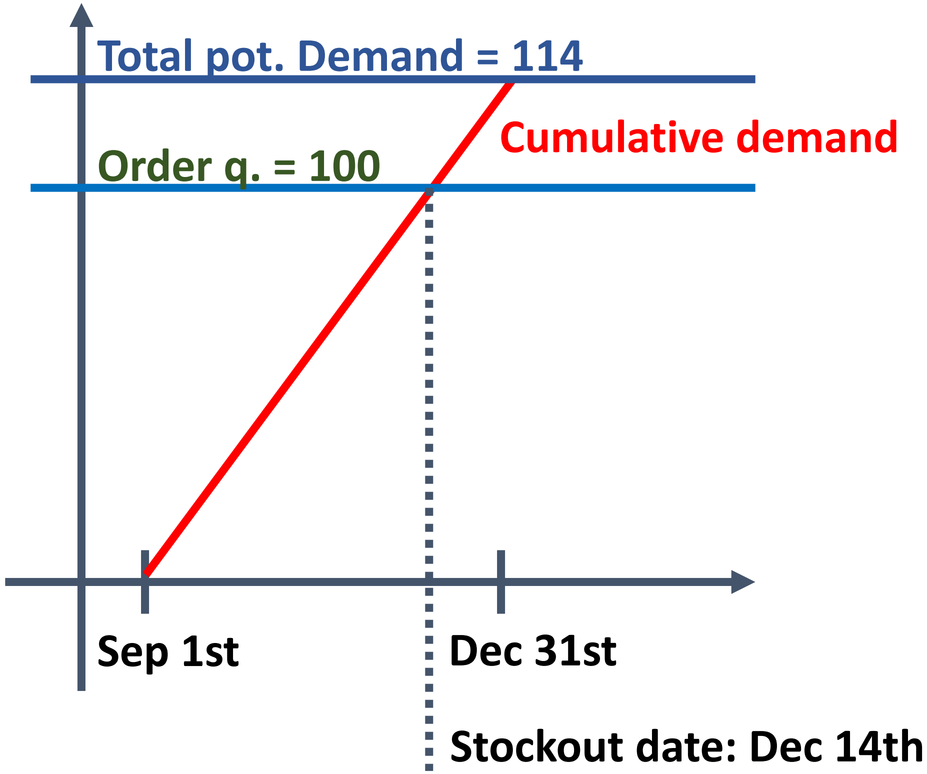
- How to be in between make-to-stock and make-to-order
- We can typically learn a lot from early demand
- Then we can place another order from a faster, but more expensive source
- This way, we can make more profit from early demand

Couple words about demand

It is difficult to keep track of lost sales
So we can only observe the demand if its less than the order quantity

Case 1: no censoring
Sales = Demand

Case 2: censoring
Sales = Order quantity



Reactive capacity: who uses it?

TIMBUK2

- Produces some bags in Indonesia and China
- Others in San Francisco

OBERMEYER
ASPEN · COLORADO

- Pioneer in reactive capacity
- Makes two orders: one earlier in the season, one later

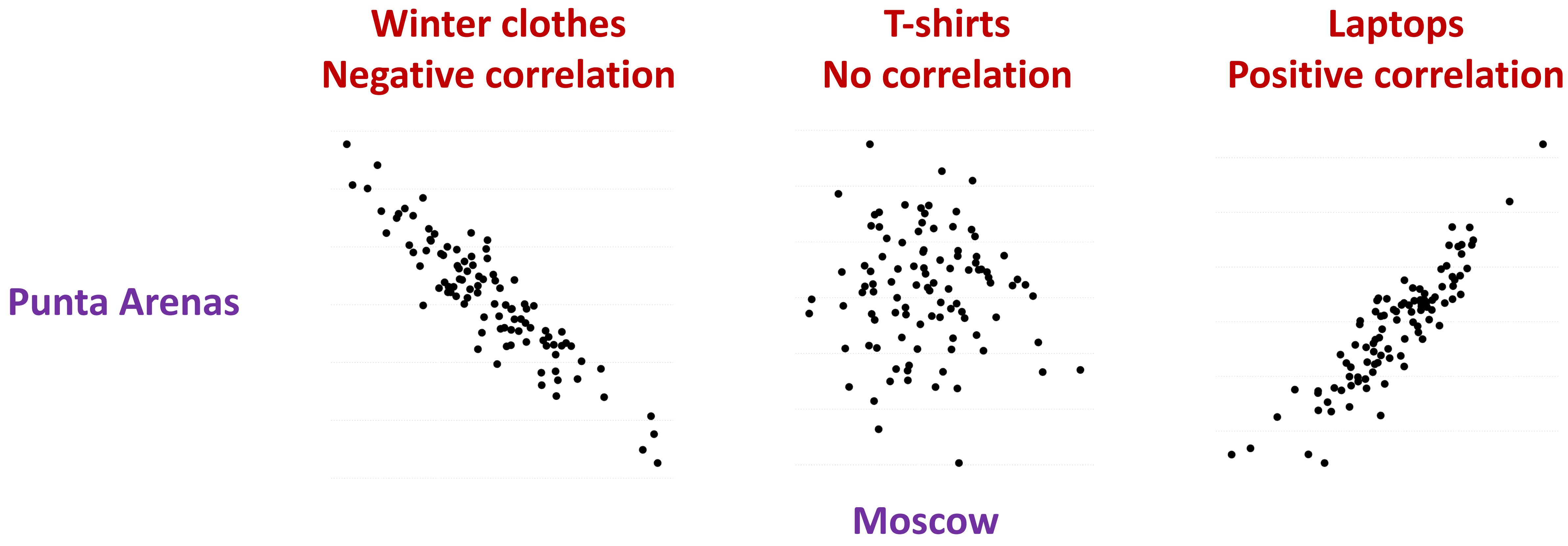

VENDE
61-222501-246522
VENDE

Demand correlation

- Two markets: Moscow and Punta Arenas
- Winter clothes: high demand in Moscow -> low in Punta Arenas
- Fashionable T-shirts: probably no dependence
- Gaming laptops: high demand in Moscow -> high in Punta Arenas

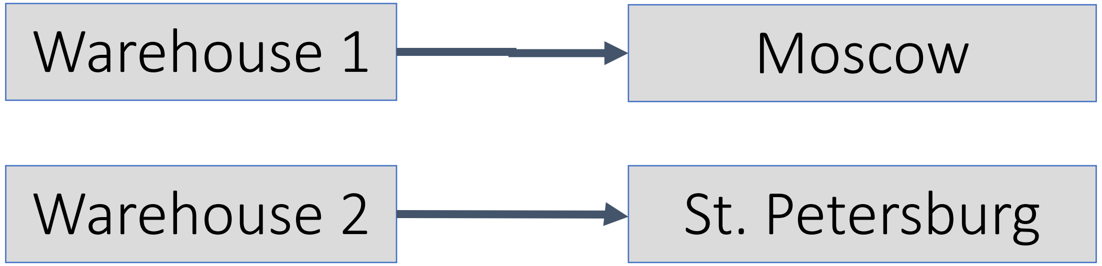
Demand correlation

- Two markets: Moscow and Punta Arenas
- Winter clothes: high demand in Moscow -> low in Punta Arenas
- Fashionable T-shirts: probably no dependence
- Gaming laptops: high demand in Moscow -> high in Punta Arenas

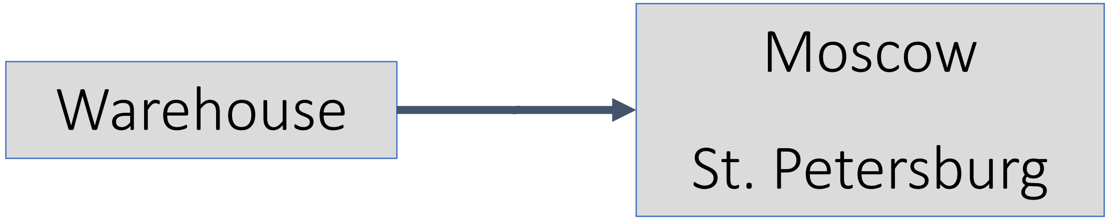


Pooling

- Two markets: Moscow and St. Petersburg
- Either use two warehouses or use one big one
- Benefit of two warehouses:
  - Cheaper and faster to deliver
- Benefit of one warehouse:
  - Reduced variability! Why?



- Moscow demand: $D_1 \sim N(\mu_1, \sigma_1)$
- St. Petersburg demand: $D_2 \sim N(\mu_2, \sigma_2)$
- Correlation: $\mathrm{Corr}(D_1, D_2) = \rho$
  - By definition, always between -1 and 1



- Moscow + St. Petersburg: $D_1 + D_2 \sim N(\mu_1 + \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2})$
- SD of the last expression is never greater than $\sigma_1 + \sigma_2$

Pooling: quick practice problem

Consider two products, A and B. Demands for both products are normally distributed and have the same mean and standard deviation. The coefficient of variation of demand for each product is 0.6. The estimated correlation in demand between the two products is −0.7. What is the coefficient of variation of the total demand of the two products?

Pooling: quick practice problem solution

Consider two products, A and B. Demands for both products are normally distributed and have the same mean and standard deviation. The coefficient of variation of demand for each product is 0.6. The estimated correlation in demand between the two products is −0.7. What is the coefficient of variation of the total demand of the two products?

Answer: (1/2) * sqrt(2 + 2 * rho) * CV = 0.23

Manufacturing flexibility

How can pooling be implemented? By making **plants** flexible.
Consider **car manufacturing**.
A **focused** plant would only produce one car model.
A more flexible one can produce multiple models.
Focus is costly because of the demand uncertainty. Flexibility incurs fixed costs.
How should we decide whether to be focused or flexible?

Plant 1	Car model A
Plant 2	Car model B
Plant 3	Car model C
Plant 4	Car model D
Plant 5	Car model E

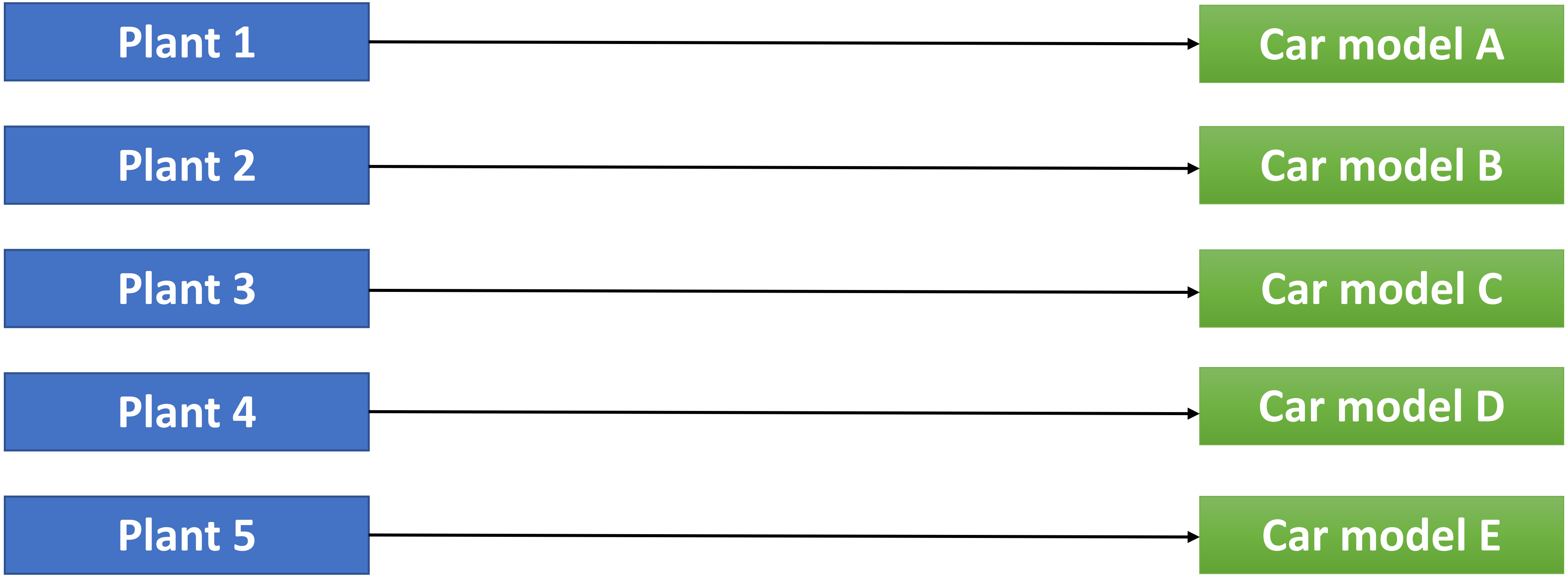
Manufacturing flexibility

Dedicated strategy

Each plant produces its own model.

Prone to demand uncertainty.

But is cheap.



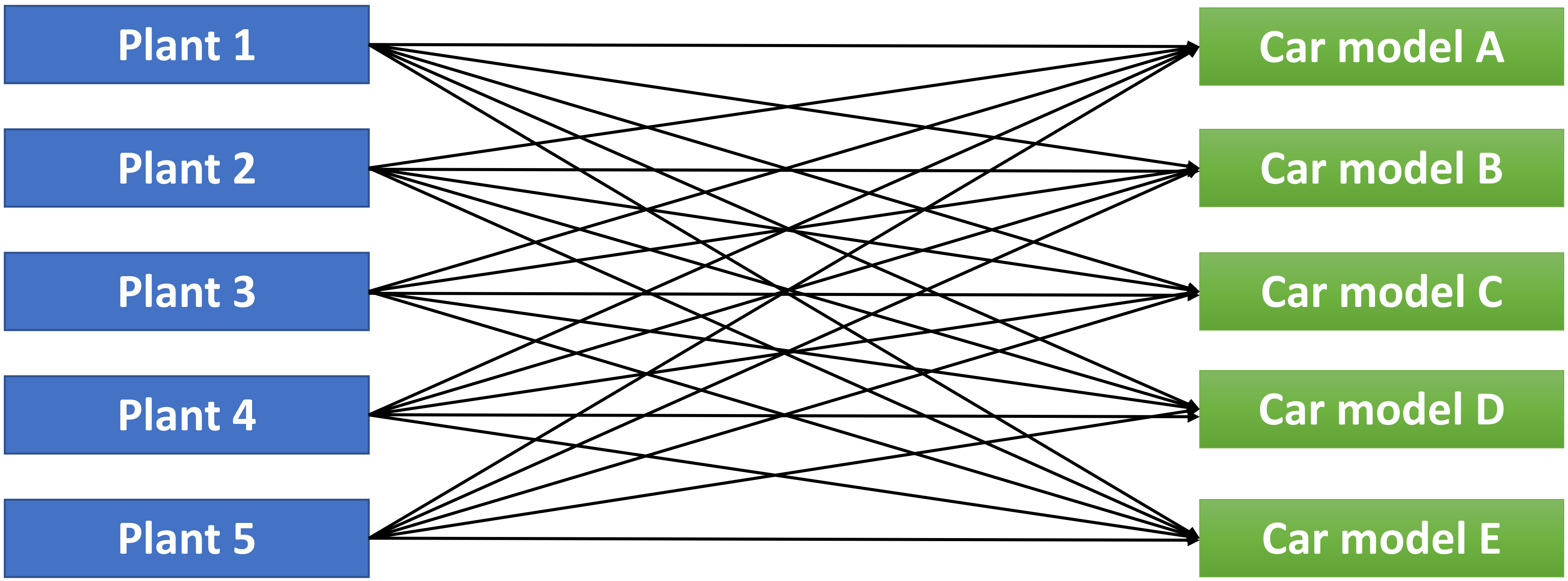
Manufacturing flexibility

Full flexibility

All plants produce all models.

Achieves complete pooling and reduces the most of demand uncertainty.

But is quite expensive, as the number of links is quadratic in the number of plants.



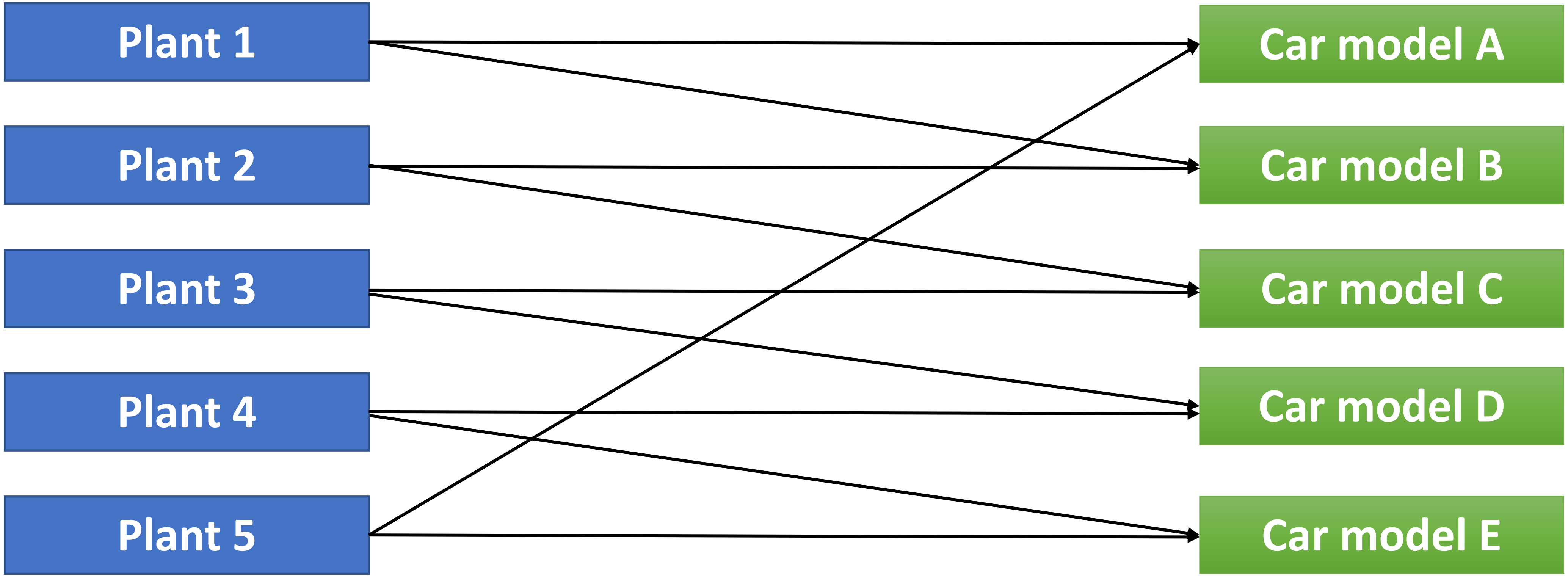
Manufacturing flexibility

Chaining

Connect all plants and products in one long cycle.

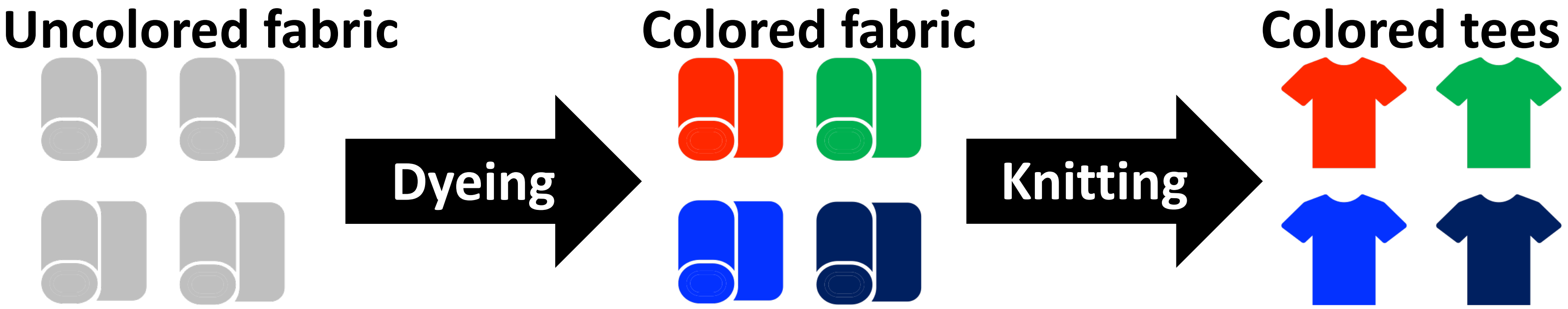
The best of both worlds.

Allows to pool most of the demand efficiently while being cheap.

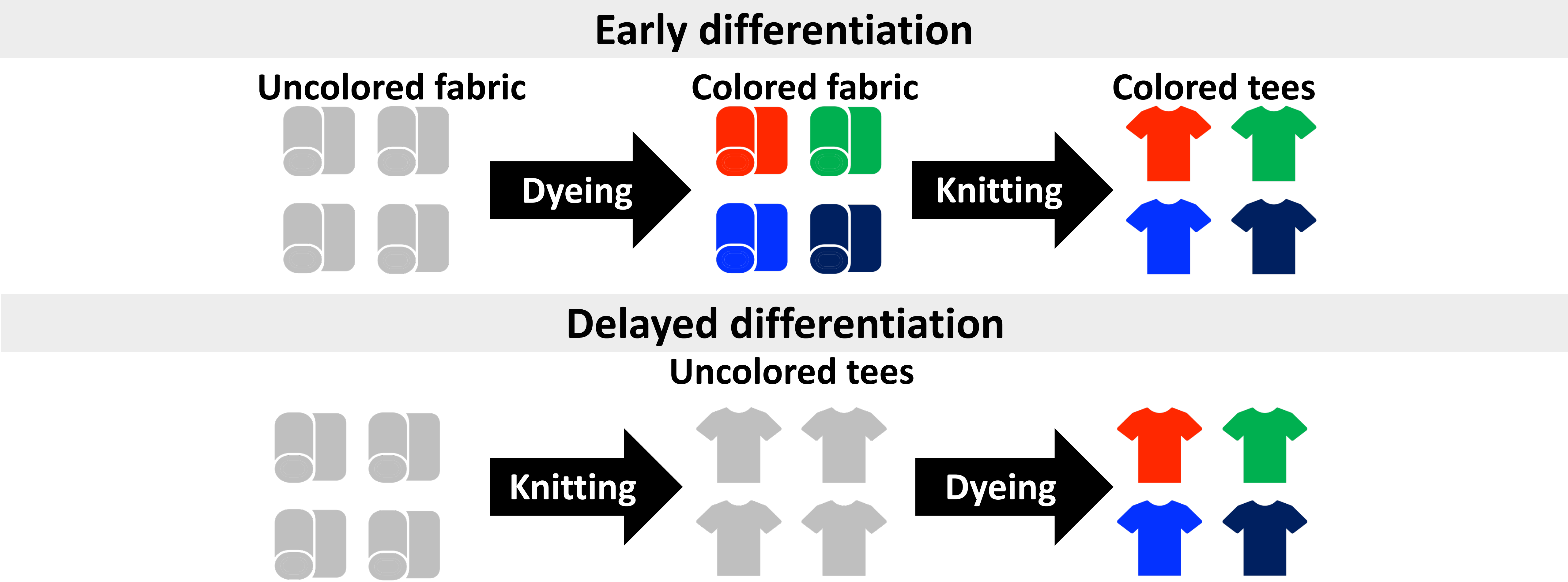


Delayed differentiation

Early differentiation

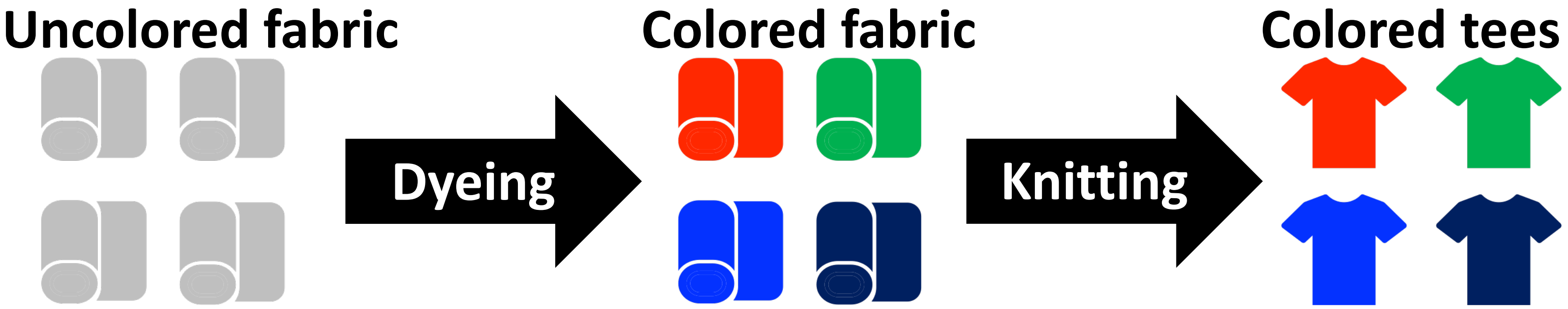


Delayed differentiation

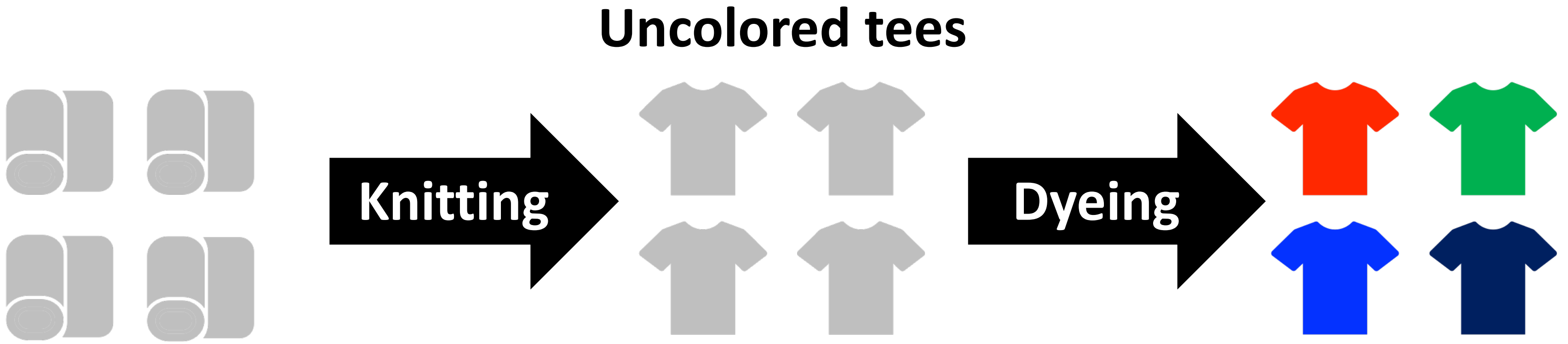


Delayed differentiation

Early differentiation



Delayed differentiation



- Delayed differentiation combines two ideas:
  - **Learning demand over time.** You can color tee-shirts mid-season (at a bit larger cost).
  - **Pooling.** You combine all different colors into one which reduces variability

Delayed differentiation: example

Early differentiation

- 10 colors, for each color:
  - Demand distributed normally: μ = 100, σ = 20
  - Demands for different colors are independent
  - Selling price p = \$200, purchase cost c = \$150, no salvage
  - Essentially 10 separate identical newsvendors

Delayed differentiation: example

Early differentiation

- 10 colors, for each color:
  - Demand distributed normally: μ = 100, σ = 20
  - Demands for different colors are independent
  - Selling price p = \$200, purchase cost c = \$150, no salvage
  - Essentially 10 separate identical newsvendors
  - Q = 86.5 for each color, exp. profit \$3 728.89 per color
  - Total profit is \$**37 288.9**.

Delayed differentiation: example

Early differentiation

- 10 colors, for each color:
  - Demand distributed normally: μ = 100, σ = 20
  - Demands for different colors are independent
  - Selling price p = \$200, purchase cost c = \$150, no salvage
  - Essentially 10 separate identical newsvendors
  - Q = 86.5 for each color, exp. profit \$3 728.89 per color
  - Total profit is \$**37 288.9**.

Delayed differentiation

- Pool the colors together (make note that demands are independent)

Delayed differentiation: example

Early differentiation

- 10 colors, for each color:
  - Demand distributed normally: μ = 100, σ = 20
  - Demands for different colors are independent
  - Selling price p = \$200, purchase cost c = \$150, no salvage
  - Essentially 10 separate identical newsvendors
  - Q = 86.5 for each color, exp. profit \$3 728.89 per color
  - Total profit is \$**37 288.9**.

Delayed differentiation

- Pool the colors together (make note that demands are independent)
- Total demand distributed normally: μ = 1000, $\sigma = 20\sqrt{10} = 63$
- Delayed coloring is more expensive: purchase cost goes up to \$155
- Q = 952.22 for uncolored tees, exp. total profit \$**41 206.37**

Delayed differentiation: who uses it

- Differently coloured sweaters
- How is differentiation delayed?
  - Sweaters are knitted colourless
  - Colour applied once the demand is clear

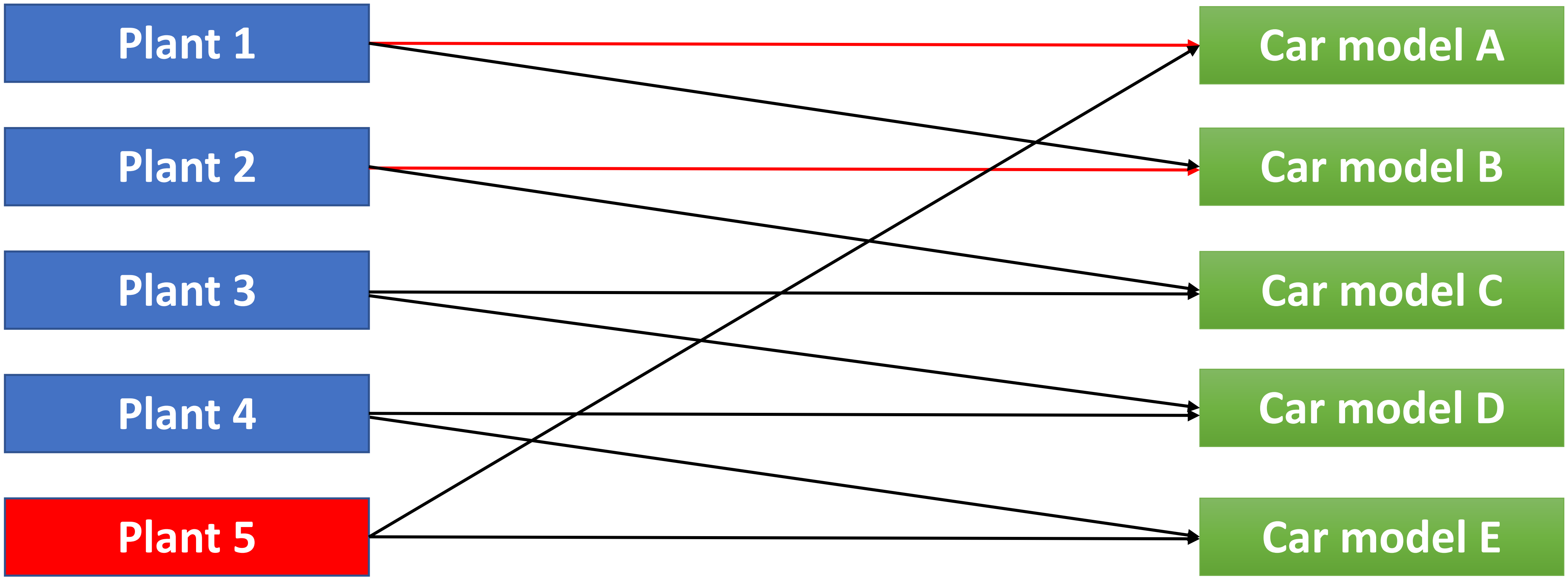
- Produces soup for private labels
- Example: supermarket brands
- How is differentiation delayed?
  - Unlabelled cans produced first
  - Labels applied once the need arises

- Printers in different countries
- Different power supply requirements
- How is differentiation delayed?
  - Generic printers
  - Power supply modules added on-demand

- Drills for different store brands
- How is differentiation delayed?
  - Drills produced with no packaging
  - Packages applied once the product is sent

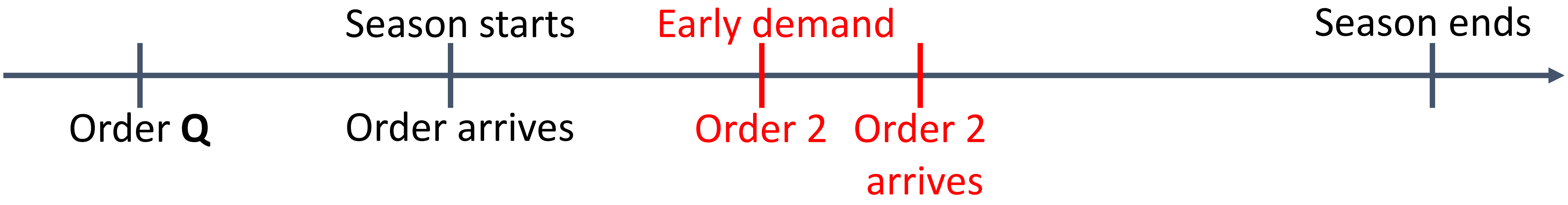
Project ideas: Flexibility with disruption

- Consider the production flexibility setting
- What if now, with some probability, some plants do go down?
  - Or it is impossible to produce a certain car at a given plant?
- Formulate a stochastic programming model, solve it for an example, and experiment!



Project ideas: RC in a supply chain

Reactive capacity



- Consider a supply chain of a slow manufacturer and a retailer
- Every company aims to maximize their expected profit
- Demand is random; there is a certain correlation between season start and season end demand
- A retailer sells the product for unit price **p,** a manufacturer produces it at unit cost **c**
- A manufacturer sells the product to the retailer at unit wholesale price **w**
- Initially, the retailer does not use reactive capacity
- If the retailer now employs a reactive capacity strategy and enters an arrangement with expensive manufacturer producing at cost **C**, will this arrangement ever be beneficial for slow manufacturer?

Project structure: recap

- 13% of the final grade (26% if you’re only taking my half)

- What will you need to do?
  - **Propose** a business problem
  - **Formalize** its solution as an optimization problem
  - **Implement** a prototype solution given some example data
  - **Explore** numerically how it behaves

- Each team submits a proposal to me by April 25^{th}
  - Only one submission per team is needed

Project structure: proposal

- Three paragraphs, **no more** than 300 words

- **Paragraph 1:** intro. Describe the problem you're solving and make the case for it being interesting.

- **Paragraph 2:** your proposed solution. Describe an optimization or simulation algorithm verbally (but briefly). You can also include the math.

- **Paragraph 3:** planned experiments. Which properties of the model would you like to explore?

- Submit a .pdf file to me by e-mail. One submission per team.

Project grading criteria: recap

- How can you gain points for the midterm project?
  - **Creativity:** does the idea stand out?
  - **Formalization:** is the model either original or technically sophisticated?
  - **Implementation:** are you able to solve the problem to optimality?
  - **Exploration:** did you evaluate your model? On how many dimensions?

Up next

- Assignment 2 due on April 21st 23:59:59
- Assignment 3 due on May 2nd 23:59:59

- Submit your project proposal by April 25th 23:59:59

- Seminar: getting comfortable with stochastic programming

- Next lectures:
  - Dynamic programming: optimizing over time
  - Applications: job search, airline pricing, etc.
  - What to study after this course?

Business analytics I
Operations Analytics
Class 7
Stochastic Dynamic Programming

Marat Salikhov
April 25th , 2022

Dynamic decisions over time

- How can we formalize **planning** under uncertainty?
- We need the following components:
  - States of the world
  - Uncertainty
  - Actions
  - Payoffs
- Markov chains represented the first two
- Now we add in actions and payoffs. This way, we get a modeling tool called **Markov decision process**

Markov decision process

- Time periods t indexed as $0, \ldots, T$
  - **Terminal** period T: nothing is done in this period

Markov decision process

- Time periods t indexed as 0, ... , T
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)

Markov decision process

- Time periods t indexed as 0, ... , T
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)
- Set of **actions** A_s (depending on a state s in S)

Markov decision process

- Time periods t indexed as $0, \ldots, T$
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)
- Set of **actions** A_s (depending on a state s in S)
- Set of **random disturbances** W_s (depending on a state s in S)
  - Probability distribution: $w_t \sim G(\cdot \mid s_t, a_t)$

Markov decision process

- Time periods t indexed as $0, \ldots, T$
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)
- Set of **actions** A_s (depending on a state s in S)
- Set of **random disturbances** W_s (depending on a state s in S)
  - Probability distribution: $w_t \sim G(\cdot \mid s_t, a_t)$
- **Instantaneous cost** in period t: $g_t(s, a, w)$ with $s \in S$, $a \in A_s$ and $w \in W_s$

Markov decision process

- Time periods t indexed as $0, \ldots, T$
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)
- Set of **actions** A_s (depending on a state s in S)
- Set of **random disturbances** W_s (depending on a state s in S)
  - Probability distribution: $w_t \sim G(\cdot \mid s_t, a_t)$
- **Instantaneous cost** in period t: $g_t(s, a, w)$ with $s \in S$, $a \in A_s$ and $w \in W_s$
- **Law of motion:** $s_{t+1} = f_t(s_t, a_t, w_t)$ for all $t < T$
  - Depends on the time period

Markov decision process

- Time periods t indexed as $0, \ldots, T$
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)
- Set of **actions** A_s (depending on a state s in S)
- Set of **random disturbances** W_s (depending on a state s in S)
  - Probability distribution: $w_t \sim G(\cdot \mid s_t, a_t)$
- **Instantaneous cost** in period t: $g_t(s, a, w)$ with $s \in S$, $a \in A_s$ and $w \in W_s$
- **Law of motion:** $s_{t+1} = f_t(s_t, a_t, w_t)$ for all $t < T$
  - Depends on the time period
- Minimizing a total cost function that is additively separable over time:

$$\min_{\mathbf{u}} \mathbb{E}_{\mathbf{w}} \left[g_T(s_T) + \sum_{t=0}^{T} g_t(x_t, u_t, w_t) \right]$$

Markov decision process

- Time periods t indexed as $0, \dots, T$
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)
- Set of **actions** A_s (depending on a state s in S)
- Set of **random disturbances** W_s (depending on a state s in S)
  - Probability distribution: $w_t \sim G(\cdot \mid s_t, a_t)$
- **Instantaneous cost** in period t: $g_t(s, a, w)$ with $s \in S$, $a \in A_s$ and $w \in W_s$
- **Law of motion:** $s_{t+1} = f_t(s_t, a_t, w_t)$ for all $t < T$
  - Depends on the time period
- Minimizing a total cost function that is additively separable over time:

$$\min_{\mathbf{u}} \mathbb{E}_{\mathbf{w}} \left[g_T(s_T) + \sum_{t=0}^{T} g_t(x_t, u_t, w_t) \right]$$

- **Policy** $\pi = \{\mu_0, \dots, \mu_{T-1}\}$
- Each μ_t maps s_t to $a_t \in A_{s_t}$
- Interpretation: plan of action

MDP example: Inventory management

- Consider the following inventory management setting
- There are multiple time periods denoted by t
- At the beginning of each period t the firm starts out with s_t units of product in inventory
- The firm then decides how many units of product a_t to produce, costing it c dollars
- Then, a random realization of demand w_t is realized, and up to $s_t + q_t$ units are sold at price r
- Finally, $\min(s_t + a_t - w_t, 0)$ units of inventory are carried over to next season

MDP example: Inventory management

- Set of **states** S: inventory values
- Set of **actions** A_s: purchase orders

MDP example: Inventory management

- Set of **states** S: inventory values
- Set of **actions** A_s: purchase orders
- Set of **random disturbances** W_s: demand values
  - Demand is independent of inventory or purchase orders: $w_t \sim H(\cdot)$

MDP example: Inventory management

- Set of **states** S: inventory values
- Set of **actions** A_s: purchase orders
- Set of **random disturbances** W_s: demand values
  - Demand is independent of inventory or purchase orders: $w_t \sim H(\cdot)$
- **Instantaneous profit** in period t: $g_t(s_t, a_t, w_t) = r \min(s_t + a_t, w_t) - ca_t$
  - Per-unit revenue r, and per-unit cost c

MDP example: Inventory management

- Set of **states** S: inventory values
- Set of **actions** A_s: purchase orders
- Set of **random disturbances** W_s: demand values
  - Demand is independent of inventory or purchase orders: $w_t \sim H(\cdot)$
- **Instantaneous profit** in period t: $g_t(s_t, a_t, w_t) = r \min(s_t + a_t, w_t) - ca_t$
  - Per-unit revenue r, and per-unit cost c
- **Law of motion:** $s_{t+1} = \min(s_t + a_t - w_t, 0)$ for all $t < T$
  - You cannot sell more than inventory at hand

MDP example: Inventory management

- Set of **states** S: inventory values
- Set of **actions** A_s: purchase orders
- Set of **random disturbances** W_s: demand values
  - Demand is independent of inventory or purchase orders: $w_t \sim H(\cdot)$
- **Instantaneous profit** in period t: $g_t(s_t, a_t, w_t) = r \min(s_t + a_t, w_t) - ca_t$
  - Per-unit revenue r, and per-unit cost c
- **Law of motion:** $s_{t+1} = \min(s_t + a_t - w_t, 0)$ for all $t < T$
  - You cannot sell more than inventory at hand
- Maximizing expected profit over time:

$$\max_{\mathbf{a}} \mathbb{E}_{\mathbf{w}} \left[g_T(s_T) + \sum_{t=0}^{T} g_t(x_t, a_t, w_t) \right]$$

MDP example: Shopping

A shopper is looking for the best price on a particular home appliance. Each time he searches, he receives a price quote randomly drawn from B(10, 0.4) distribution. His estimated costs per each search are $1. A shopper is willing to go through at most 20 alternatives before giving up and going with the currently best one. He can also terminate the search early and buy the currently best alternative. What is the shopper's optimal value?

MDP example: Shopping

- Set of **states** S: pairs (k, l). k is curr. best price, l is 1 if app. not bought yet
- Set of **actions** A_s: buy now (0) or shop for more (1). Shopping is only possible if $l = 1$: $a_t \leq l_t$.
- Set of **random disturbances** W_s: price opportunities
  - Binomially distributed: $w_t \sim B(n, p)$
- **Instantaneous cost** in period $t < T$: $g_t(k_t, l_t, a_t, w_t) = ca_t$
  - Search cost c
- **Terminal cost** in period T: $g_T(k_T, l_T, a_T, w_T) = k_T$
- **Law of motion:**
  - You either buy or shop: $k_{t+1} = (1 - a_t)k_t + a_t \min(k_t, w_t)$
  - If you buy, you can no longer shop: $l_{t+1} = a_t$
- Minimizing expected cost over time:

$$\min_{\mathbf{a}} \mathbb{E}_{\mathbf{w}} \left[g_T(s_T) + \sum_{t=0}^{T} g_t(x_t, a_t, w_t) \right]$$

Finding optimal policies

- Time periods t indexed as $0, \ldots, T$
  - **Terminal** period T: nothing is done in this period
- Set of **states** S (inventory, stock price, etc...)
- Set of **actions** A_s (depending on a state s in S)
- Set of **random disturbances** W_s (depending on a state s in S)
  - Probability distribution: $w_t \sim G(\cdot \mid s_t, a_t)$
- **Instantaneous cost** in period t: $g_t(s, a, w)$ with $s \in S$, $a \in A_s$ and $w \in W_s$
- **Law of motion:** $s_{t+1} = f_t(s_t, a_t, w_t)$ for all $t < T$
  - Depends on the time period
- Minimizing a total cost function that is additively separable over time:

$$\min_{\mathbf{u}} \mathbb{E}_{\mathbf{w}} \left[g_T(s_T) + \sum_{t=0}^{T} g_t(x_t, u_t, w_t) \right]$$

- **Policy** $\pi = \{\mu_0, \ldots, \mu_{T-1}\}$
- Each μ_t maps s_t to $a_t \in A_{s_t}$
- Interpretation: plan of action

- Expected cost of policy starting at s_0: $J_\pi(s_0) = \mathbb{E}\left[g_T(s_T) + \sum_t^{T-1} g_t(s_t, \mu_t(s_t), w_t)\right]$
- Optimal cost function: $J^*(s_0) = \min_\pi J_\pi(s_0)$
- Optimal policy π^* satisfies: $J_{\pi^*}(s_0) = J^*(s_0)$

Bellman's principle of optimality

- **Solve the problem from the end!**
- Let $\pi^* = \mu_0^*, \mu_1^*, \dots, \mu_{T-1}^*$ be an optimal policy
- Consider the *tail subproblem*:

$$\min_{\pi_i} \mathbb{E}\left[g_T(s_T) + \sum_{\tau=t}^{T-1} g_\tau(s_\tau, \mu_\tau(s_\tau), w_\tau)\right]$$

and the *tail policy*:

$$\pi_i = \{\mu_t^*, \mu_{t+1}^*, \dots, \mu_{T-1}^*\}$$

- **Principle of optimality:** The tail policy is optimal for the tail subproblem.
- Idea for the algorithm: solve all tail subproblems of a given length using the solutions of smaller subproblems

Finite-horizon dynamic programming algorithm

- Start with $J_T(s_T) = g_T(s_T)$
- Go backwards:

$$J_t(s_t) = \min_{a_t \in A_t(s_t)} \mathbb{E}\left[g_t(s_t, a_t, w_t) + J_{t+1}(f_t(s_t, a_t, w_t))\right]$$

- Then $J_0(s_0)$ is equal to the optimal cost $J^*(s_0)$, and the induced policy is optimal

Application: revenue management

Economy seats—not the same price

Some are refundable
Some require return tickets

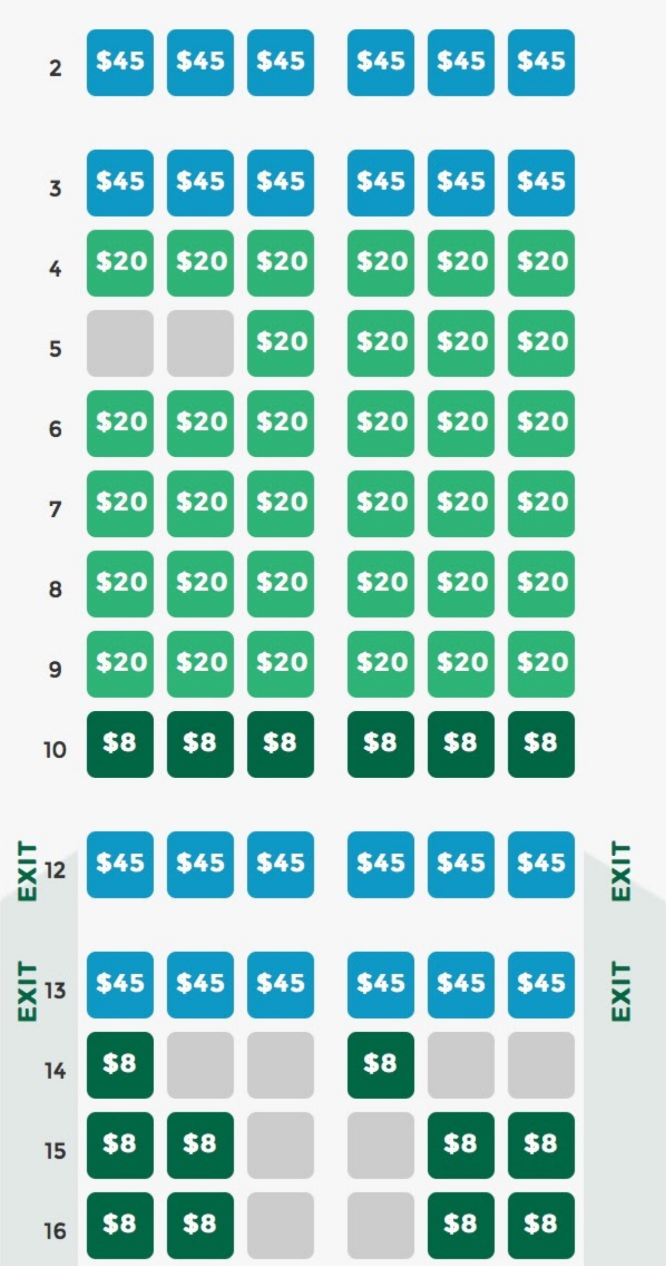
Example: Q and Y fares
Qs are more expensive than Y

Leisure vs. business

- Leisure book earlier, care about price
- Don’t care much about exact trip times
- Sunday night stayover works for them
- Notice that it is all about **economy** seats

Additionally

- Govt. employees
- Senior citizens
- Children



How to split between two fare classes?

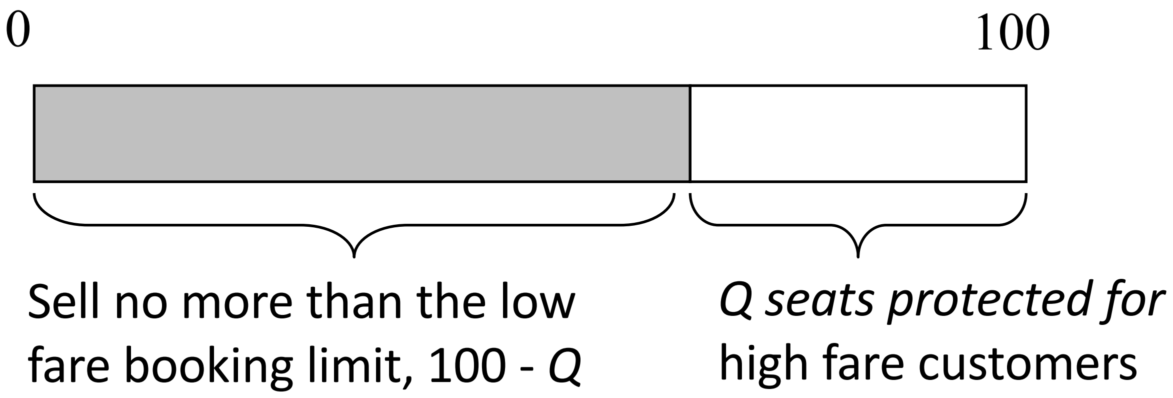
Very important question!

Am. Air. CEO circa 1985:

“We estimate that revenue management has generated **$1.4 billion in incremental revenue** in the last three years. “

Booking limits and protection classes

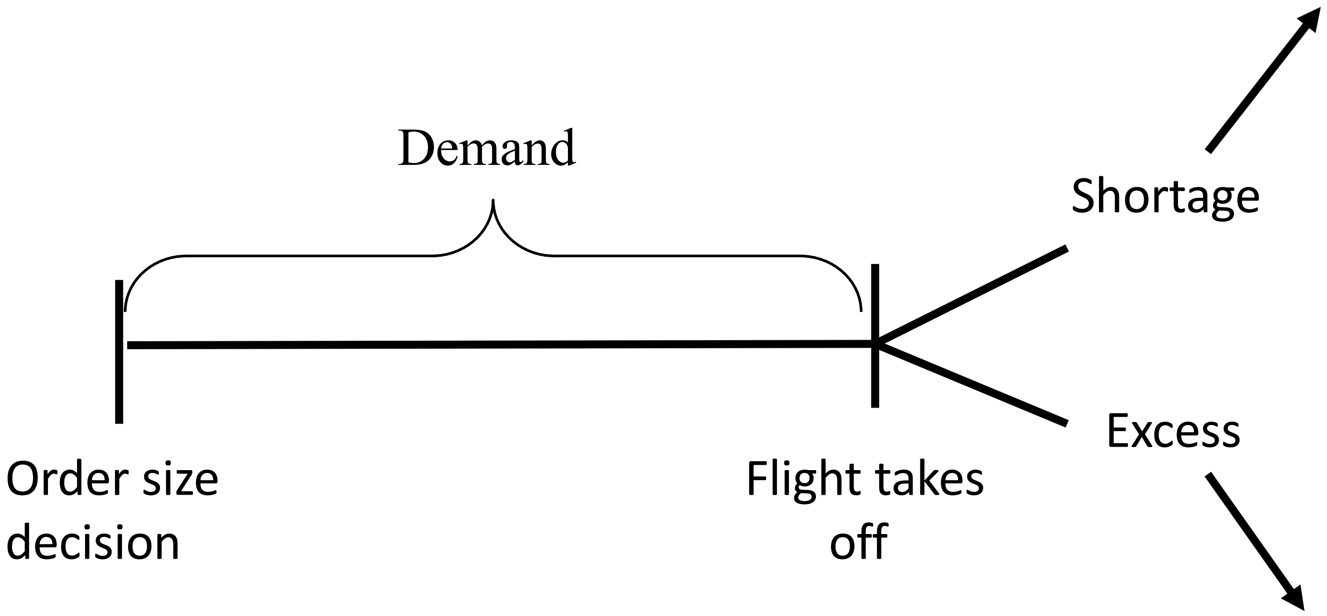
- The ***booking limit*** is the number of airline seats you are willing to sell in a fare class or lower.
- The ***protection level*** is the number of airline seats you reserve for a fare class or higher.
- Let Q be the protection level for the high fare class.
- *Q is in effect while you sell low fare tickets.*
- *Say you have 100 seats in total*
- With two fare classes, the booking limit on the low fare class is $100 - Q$:
  - You will sell no more than $100 - Q$ low fare seats because you are protecting (or reserving) Q seats for high fare customers.



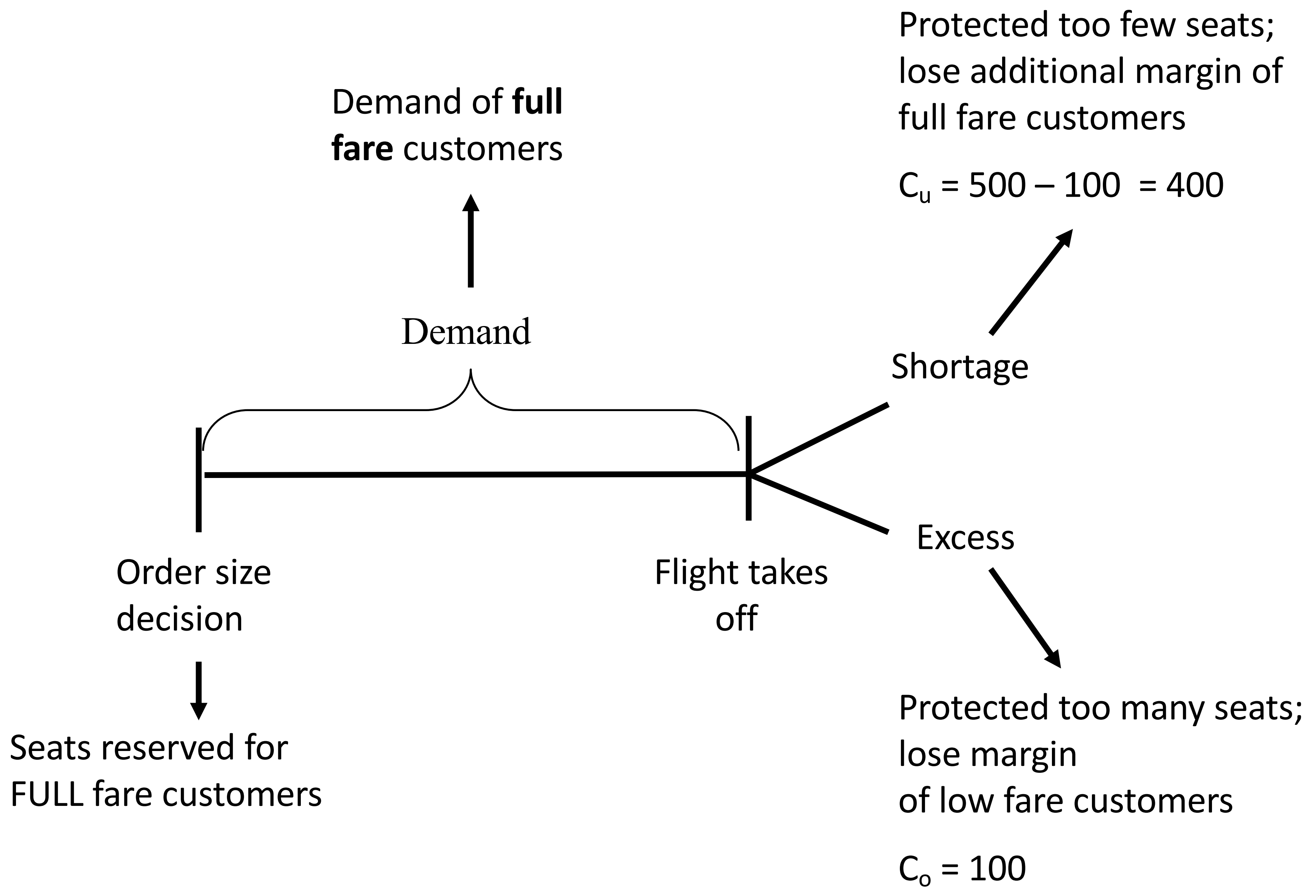
Example problem

An aircraft has 100 seats, and there are two types of fares: full ($500) and discount ($100). Although there is unlimited demand for the discount fare, demand for full fare is estimated to be equally likely anywhere between 11 and 30. How many seats should be protected for full-fare passengers booking at the last minute?

Example problem



Example problem



Example problem: dynamic program

- Two time periods: T = 2
- Set of **states** S: seats available to high-fare
- Set of **actions** A_s: how much to sell to low-fare $a_0 \leq s_0, a_1 = 0$
- Set of **random disturbances** W_s: demand for high-fare seats
  - $W_0 \equiv 0$, $W_1 \sim U(11, 30)$
- **Instantaneous profit**
  - t = 0: $g_0(s_0, a_0, w_0) = r_L a_0$
  - t = 1: $g_1(s_1, a_1, w_1) = r_H \min(s_1, w_1)$
  - $r_L = \$100$, $r_H = \$500$
- **Terminal profit** in period T: $g_T(s_T) = 0$ (The plane departs.)
- **Law of motion:**
  - You protect the seats for high-fare: $s_1 = s_0 - a_0$
- Maximizing expected profit over time:

$$\max_{\mathbf{a}} \mathbb{E}_{\mathbf{w}} \left[g_T(s_T) + \sum_{t=0}^{T} g_t(x_t, a_t, w_t) \right]$$

- Solution: $a_0 = 26$.

```
r_l, r_h = 100, 500
s_0 = 100
probs = [ss.randint(11, 31).pmf(k + 11) for k
s_1 = cp.Variable(integer = True)
a_0 = cp.Variable(integer = True)
z = cp.Variable(20, integer = True)

cons = [s_1 >= 0, a_0 >= 0, a_0 <= s_0, z >=
for w in range(20):
    cons.append(z[w] <= 11 + w)
    cons.append(z[w] <= s_1)

objective = cp.Maximize(r_l*a_0 + cp.sum([pr

rm_prob = cp.Problem(objective, cons)
rm_prob.solve()
print(s_0 - a_0.value)
```

✓ 0.6s

```
26.0
```

Revenue management challenges

- **Demand forecasting.**
  - Seasonality, special event, changing fares.
  - Customer sell up, switch flights, wait for cheaper ticket
- **Dynamic decisions.**

- **Group reservations.**

- **Multi-leg passengers/multi-day reservations for cars and hotels**:
  - Some customers can be more valuable as they bring in more legs at once

- How to construct good “fences” to differentiate among customers?
  - One-way vs round-trip tickets.
  - Saturday-night stay requirement.
  - Non-refundability.
  - Advanced purchase requirements.

Other applications of MDPs

- **Energy management**
  - which energy sources to use
  - how much to store
  - how to handle demand changes

- **Logistics**
  - Routing cars
  - Facility location

- **Platforms**
  - Customized recommendations
  - Matching drivers to riders

- **Artificial intelligence**
  - Reinforcement learning
  - Example: DeepMind

Clarification on projects and proposals

- **Project proposals are not binding**
  - You can change plans after you did it
  - You can submit multiple proposals if you are undecided on what works best
  - The goal is to allow me to provide early feedback
  - You will be guaranteed a minimum 25/100 points for your project
    - If you submit a **reasonable** proposal for me by the deadline

- **Projects**
  - Presentation: May 14th, 14:40
  - You will have 15 minutes in total (including Q/A). Plan 10 minutes for your talk
  - Submit the report to me by e-mail on May 14^{th} by 14:00
    - Your python notebook
    - A .pdf file with your results (print your notebook or save as .pdf)
  - Notebook must contain: intro, solution on an example, experiments
    - With descriptions

Up next

- Assignment 3 due on May 2^{nd} 23:59:59
- Assignment 4 due on May 12^{th} 23:59:59

- Submit your project proposal by today. Project presentations: May 14^{th}.

- Seminar on Thursday
  - wrap up stochastic programming,
  - more problems on dynamic programming

- Lecture on Thursday
  - A recap of the course
  - What to learn next

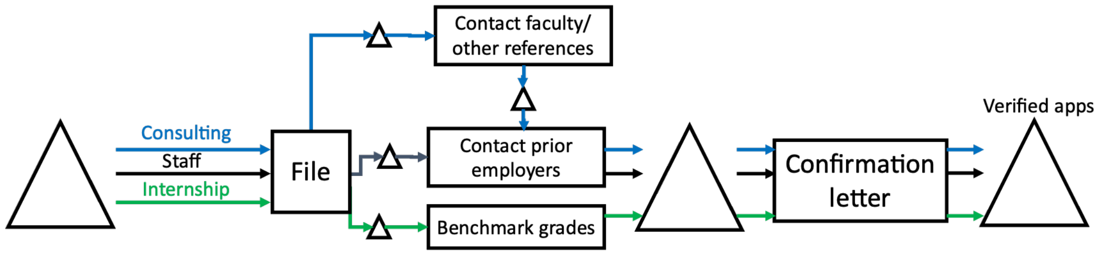
- I will put in all the information about projects in the syllabus.

Business analytics I
Operations Analytics
Class 8
Next steps

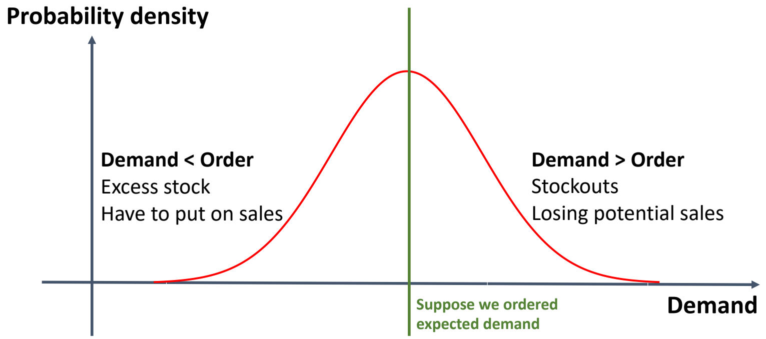
Marat Salikhov
April 28th , 2022

What we covered

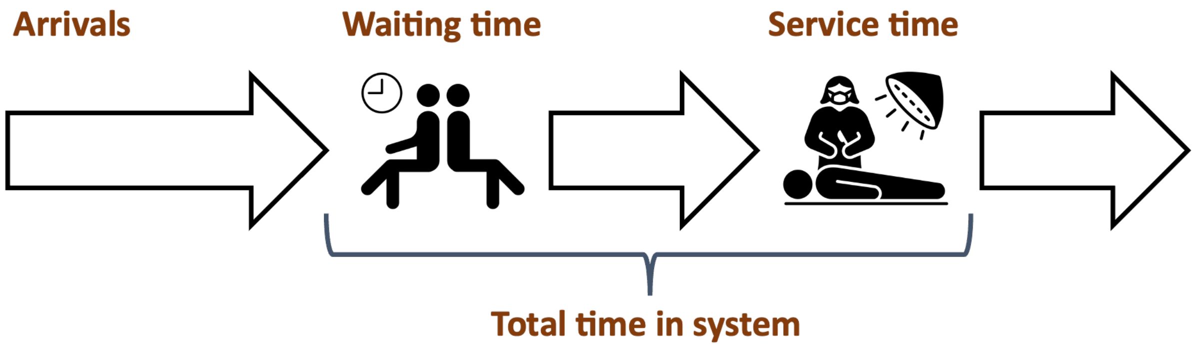
Linear programming



Stochastic optimization



Stochastic simulation



Dynamic optimization

Why did we cover it

- Modeling

- Computation skills

Couple words about projects

- Goal: encourage you to explore and “play” with things

- Think of them as Assignment 5

- Should be approximately close in difficulty to HAs
  - But involve new ideas

- You get more points for them since you come up with the questions

- Don’t try to put a lot of things at once
  - Be sure to have a single non-trivial idea
  - Add it, see what happens, and if you have spare time, add more

What we didn’t cover

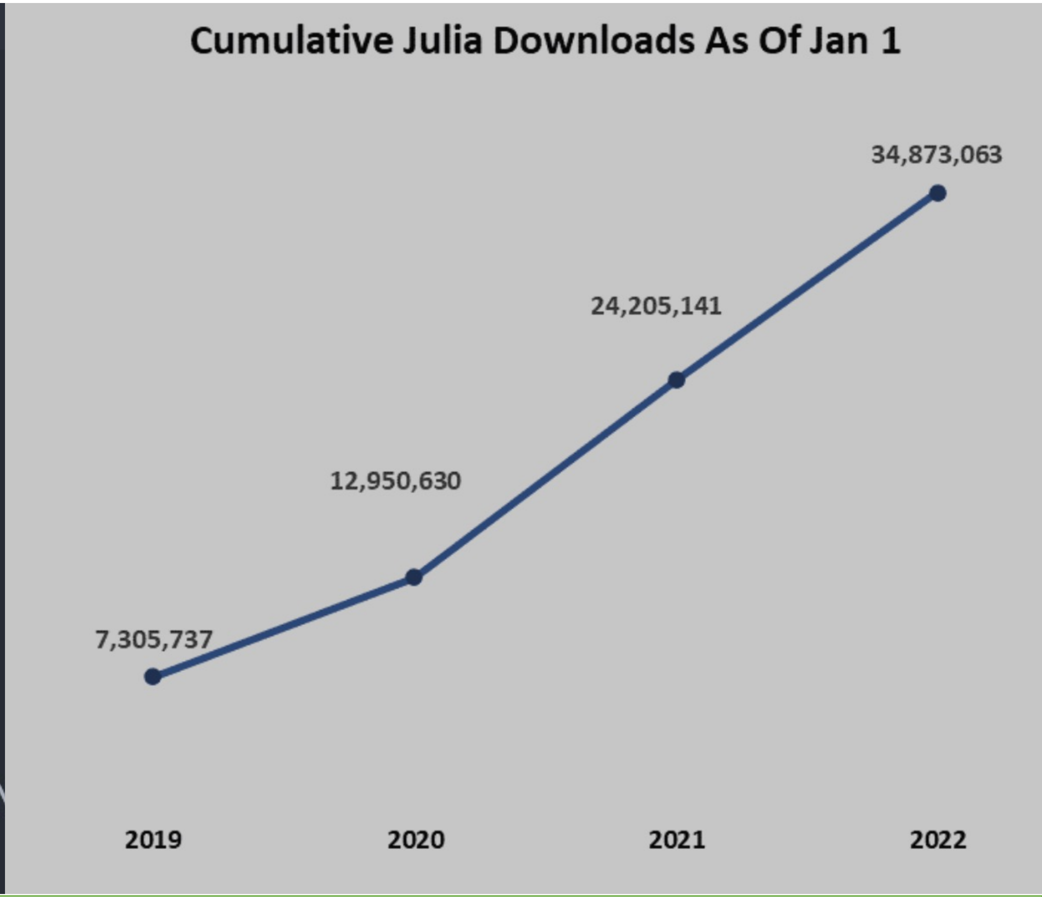
- Modeling
- Computation skills
- **Scalability**
- **Theory**
- **Implementation**

Julia is the best language for optimization

```julia
using JuMP
using SCIP

N = 13
model = Model(SCIP.Optimizer)
@variable(model, ess[1:N, 1:N], Bin)
@variable(model, eta[1:N, 1:N], Bin)
@variable(model, S[1:N], Int)
@objective(model, Min, 0)

for k in 1:N
    @constraint(model, 0 <= S[k] <= 100)
end
for k in 2:(N-1)
    @NLconstraint(model, 2*S[k] == sum(ess[k, j]*S[j] for j in 1:N
end
```



Benefits: very fast, great support for optimization, most researchers use it

Drawbacks: not as popular as Python, limited data analysis support, still a bit immature

cvxpy can solve nonlinear problems too!

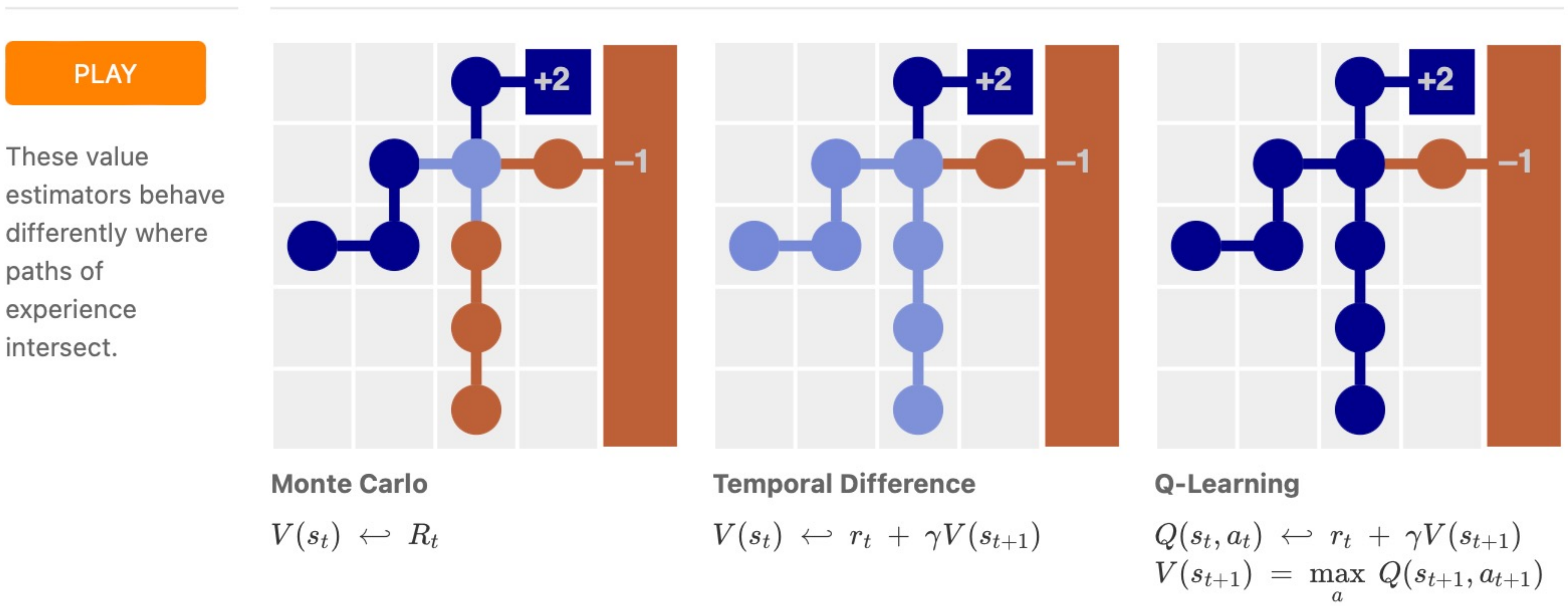
```
# Construct a CVXPY problem
x = cp.Variable(n, integer=True)
objective = cp.Minimize(cp.sum_squares(A @ x - b))
prob = cp.Problem(objective)
prob.solve()
```

Major application of cvxpy: portfolio optimization.

MDPs in reinforcement learning

Most Markov decision problems are extremely high-dimensional.
Cannot solve them exactly: need to approximate.
Reinforcement learning (teaching agents how to play) essentially does this

https://distill.pub/2019/paths-perspective-on-value-learning/



http://incompleteideas.net/book/RLbook2020.pdf

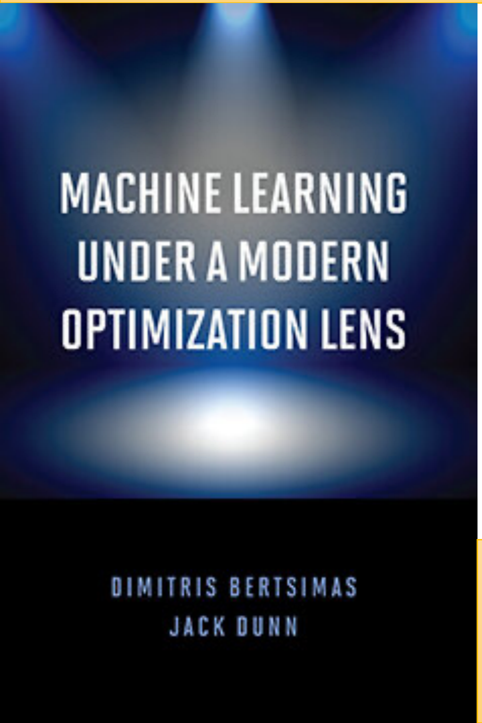
MIP in interpretable AI

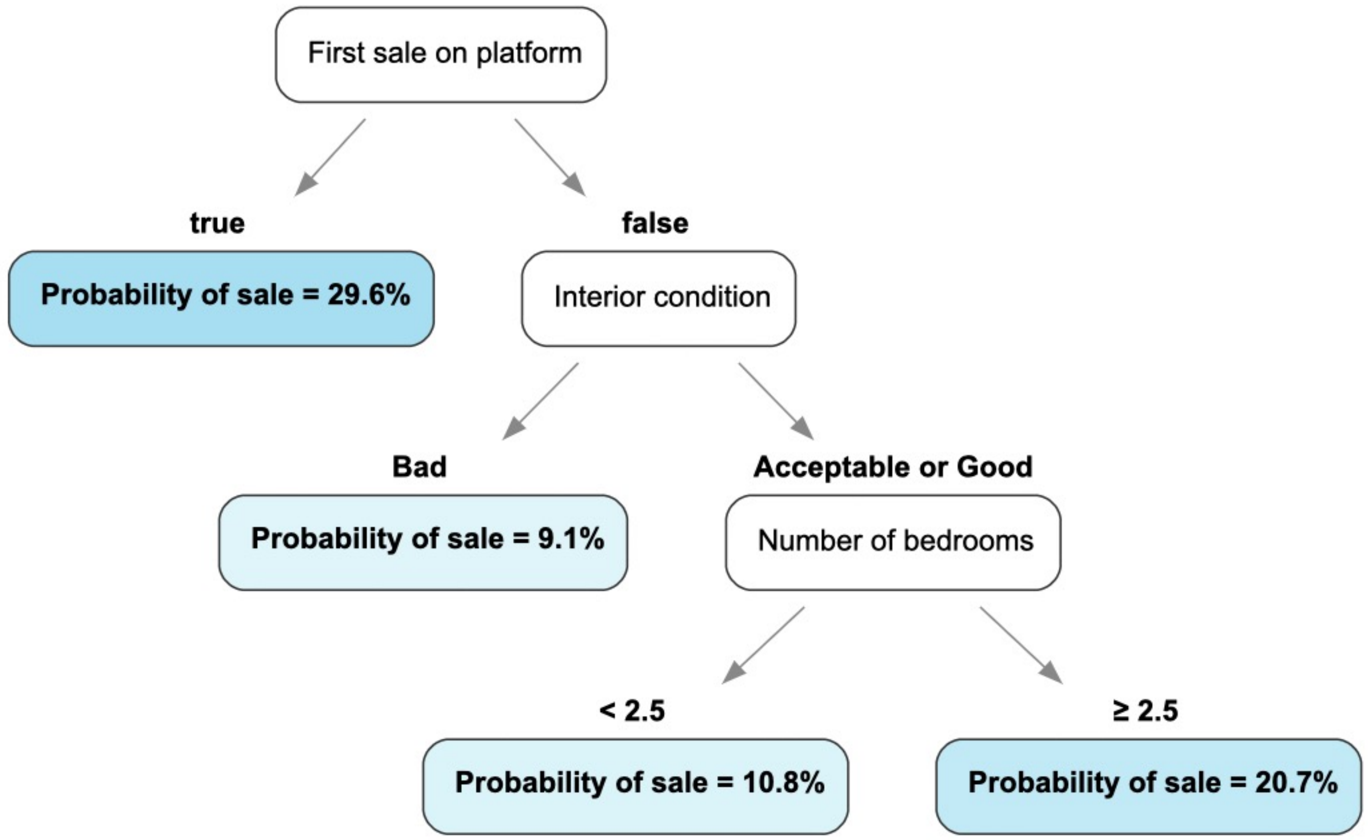
Decision models must be interpretable.

Trees are the best at that.

Existing methods were
heuristic and suboptimal.

Recent advances in MIP:
optimal decision trees





Example Optimal Decision Tree predicting the probability that a property is going to sell on a real estate platform

Implementations based on powerful solvers like Gurobi and CPLEX.

Macroeconomics and quant-econ

https://python.quantecon.org

Lectures
QuantEcon hosts lecture series on economics, finance, econometrics and data science.

QuantEcon.py
A high performance, open source Python code library for economics

QuantEcon.jl
A high performance, open source Julia code library for economics

Hardcopy Text
A hardcopy text on Quantitative Economics is currently in progress.

Workshops and Seminars
Previous and forthcoming workshops and seminars with accompanying resources

Jupyter Book
QuantEcon a founding member of the Executable Books Project, which develops Jupyter Book.

QE Notes
An open Jupyter notebook library for economics and finance

Cheatsheets
Scientific programming cheatsheets for Python and Julia

Notebooks
Browse, preview and download QuantEcon notebooks

24.3. Marginal Distributions

Now let's look at the marginal distribution ψ_T of X_T for some fixed T.

We will do this by generating many draws of X_T given initial condition X_0.

With these draws of X_T we can build up a picture of its distribution ψ_T.

Here's one visualization, with $T = 50$.

```
T = 50
M = 200  # Number of draws

ymin, ymax = 0, S + 10

fig, axes = plt.subplots(1, 2, figsize=(11, 6))

for ax in axes:
    ax.grid(alpha=0.4)
```

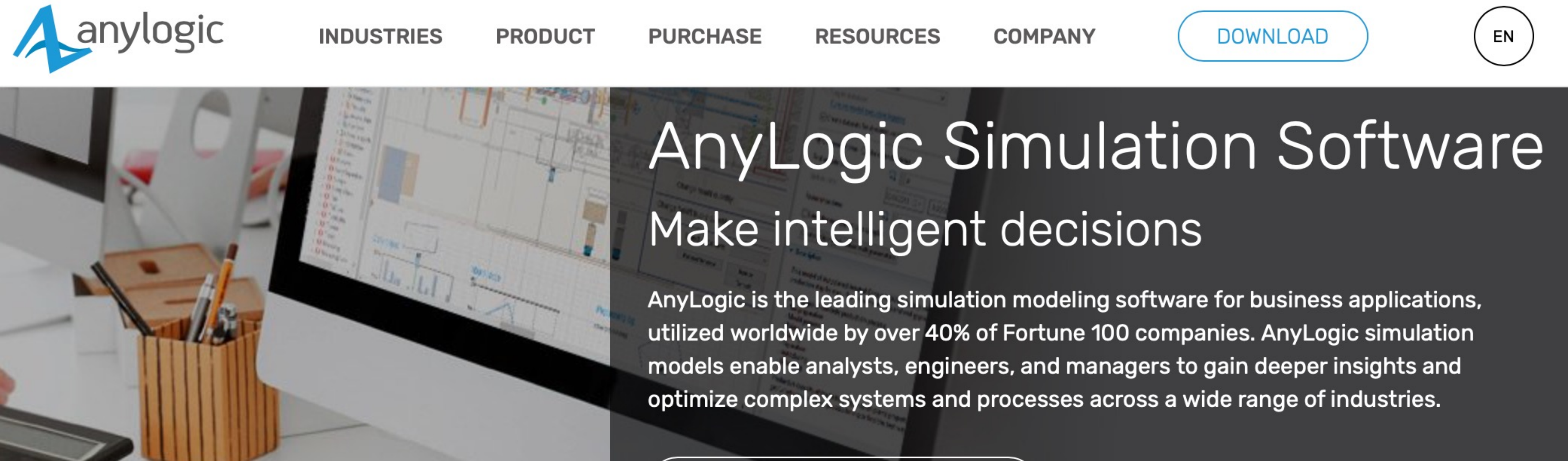
64.1. Overview

This lecture describes the concept of Markov perfect equilibrium.

Markov perfect equilibrium is a key notion for analyzing economic problems involving dynamic strategic interaction, and a cornerstone of applied game theory.

Discrete event sims: AnyLogic, simpy, etc…

https://www.anylogic.com

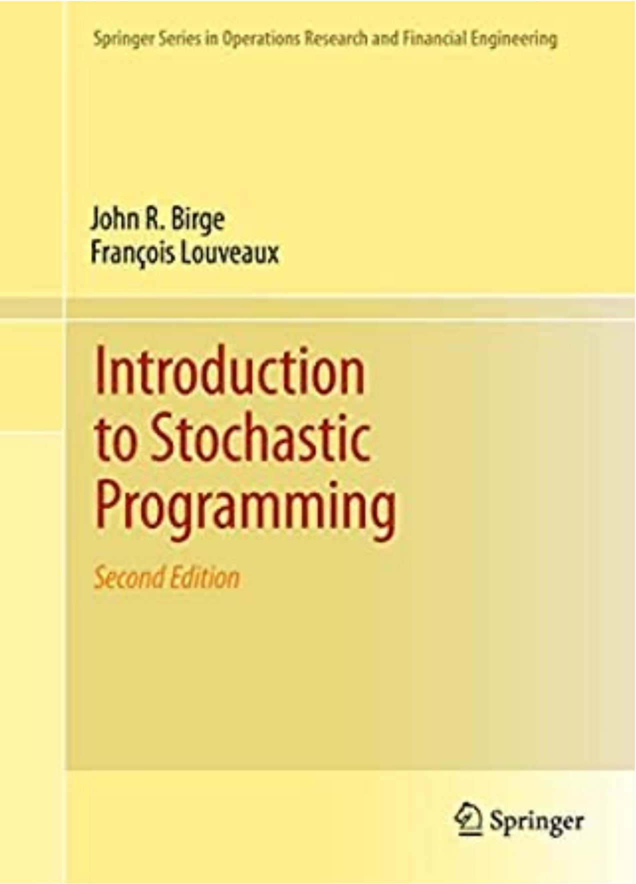


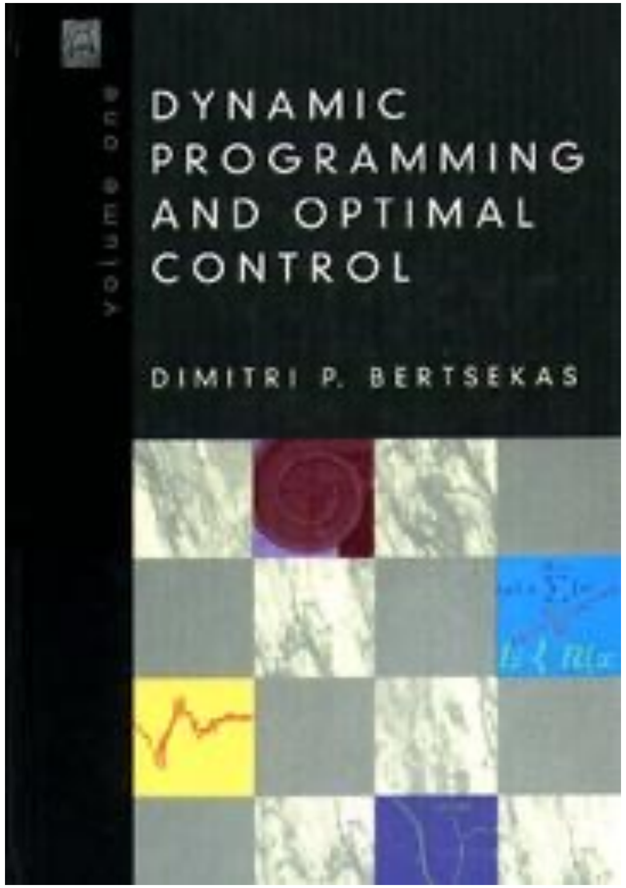
https://simpy.readthedocs.io/en/latest/

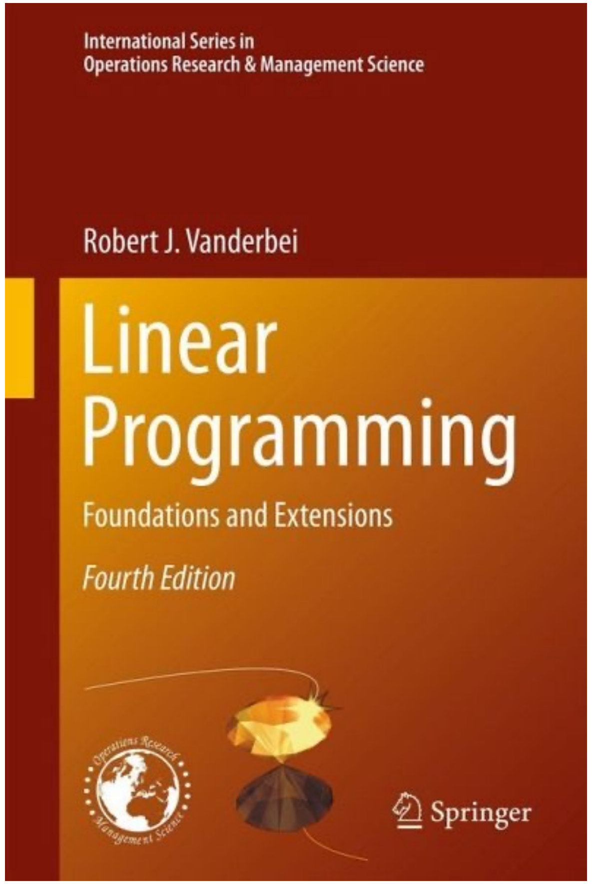
A Python library for event simulation.

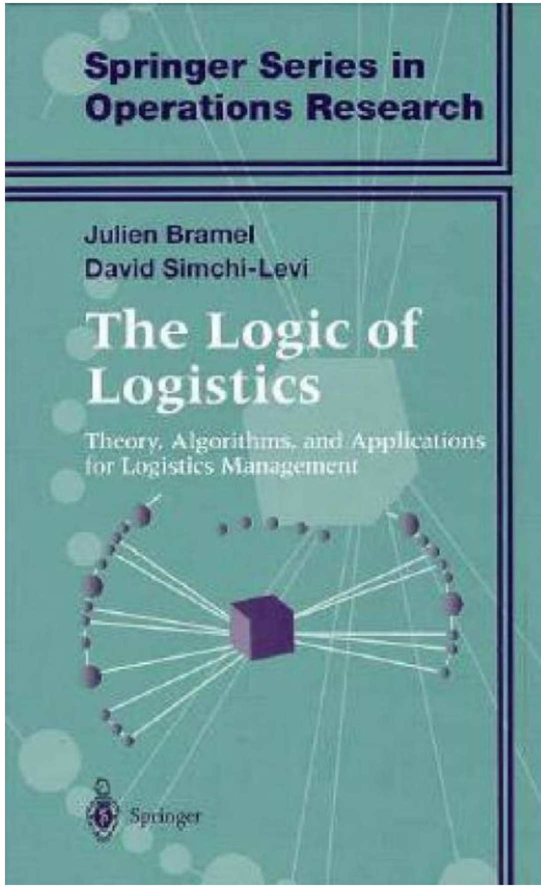
Theory and scaling

https://ocw.mit.edu/courses/6-251j-introduction-to-mathematical-programming-fall-2009/pages/lecture-notes/



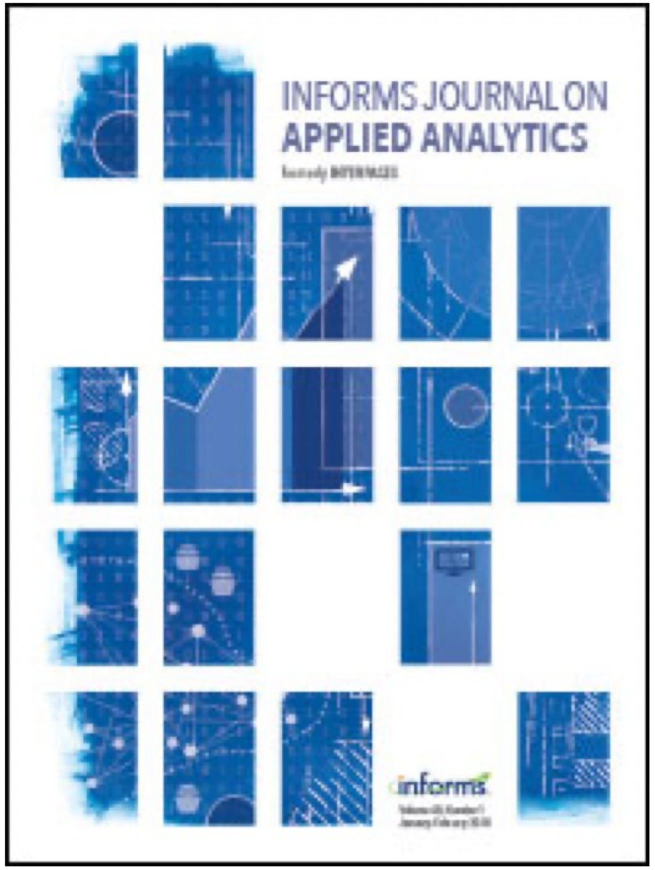






Interfaces (IJAA): examples of implementations

https://pubsonline.informs.org/journal/ijaa



INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Ride-Hailing Order Dispatching at DiDi via Reinforcement Learning

Zhiwei (Tony) Qin, Xiaocheng Tang, Yan Jiao, Fan Zhang, Zhe Xu, Hongtu Zhu, Jieping Ye

Plenty of academically reviewed papers describing how optimization is applied in practice.

TutORials in Operations Research

https://pubsonline.informs.org/series/educ

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2005

 | ISBN 1-877640-21-2
DOI 10.1287/educ.1053.0019

An Introduction to Revenue Management

Garrett J. van Ryzin
Graduate School of Business, Columbia University, 3022 Broadway, New York, New York 10027, gjv1@columbia.edu

Kalyan T. Talluri
Department of Economics and Business, Universitat Pompeu Fabra, Jaume I Building, Ramon Trias Fargas, 25–27, 08005 Barcelona, Spain, kalyan.talluri@upf.edu

Abstract *Revenue management* (RM) refers to the collection of strategies and tactics firms use to scientifically manage demand for their products and services. It has gained attention recently as one of the most successful application areas of operations research (OR).

If you would like to learn something about a particular topic in operations,
try googling “tutorials in operations research revenue management” (for example)

Up next

- Assignment 3 due on May 2nd 23:59:59
- Assignment 4 due on May 12th 23:59:59

- Online presentations on May 14th at 14:40

- Project reports due by 14:00 on May 14th

- I will get back with feedback on your proposal by the beginning of the next week

- Second half: Marketing applications
  - Most likely: lots of data analysis

- Good luck!