IO543: Scheduling

Lecture 8

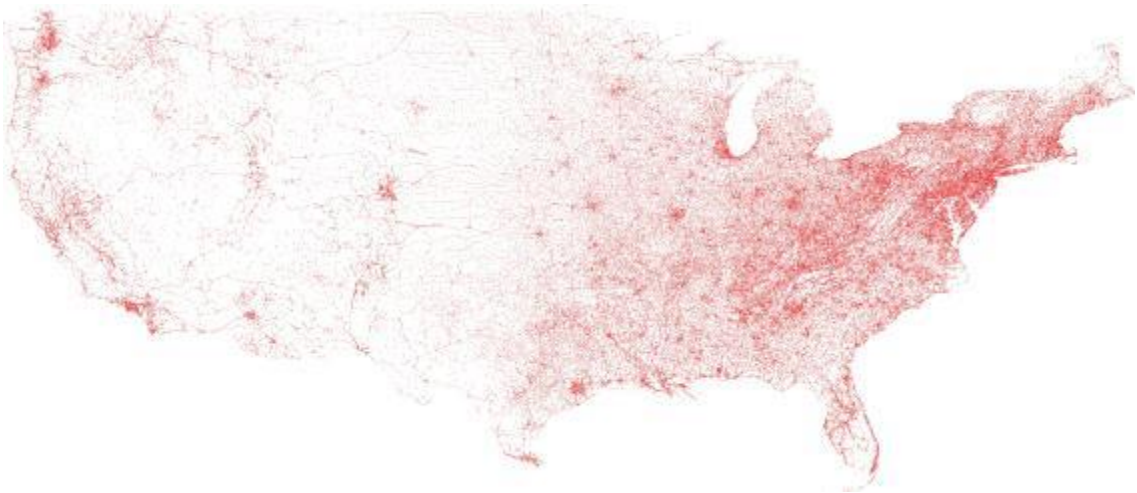
Brian Denton
Associate Professor, IOE

IOE Bld 2893
btdenton@umich.edu

Last Time

Last time we discussed the famous TSP in the context of $1 \mid s_{jk} \mid C_{max}$

Competition: $500 first prize for finding the shortest tour of the 115, 475 cities in the U.S.



Learn more at: http://www.tsp.gatech.edu/

http://www.ams.org/samplings/feature-column/fcarc-tsp

Refresher

How do you solve the following problems?

- $1 \mid \mid L_{max}$
- $1 \mid r_j \mid L_{max}$
- $1 \mid \mid \sum_j C_j$
- $1 \mid \mid \sum_j w_j C_j$
- $1 \mid prec \mid \sum_j w_j C_j$
- $1 \mid \mid \sum_j U_j$
- $1 \mid \mid \sum_j T_j$
- $1 \mid \mid \sum_j w_j T_j$
- $1 \mid s_{jk} \mid C_{max}$

Bi-Criteria Formulations

Assume there are **two criteria**, $f_1(S), f_2(S)$, for any schedule S from the set of possible schedules, θ

Possible Formulations:

$$\min_{S\in\theta} f_1(S)$$

$$\min_{S\in\theta} f_2(S)$$

$$\min_{S\in\theta}\{\alpha f_1(S) + (1-\alpha) f_2(S)\}, \quad \alpha \in [0,1]$$

$$\min_{S\in\theta}\{f_1(S) |\ s.t.\ f_2(S) \le \beta\}$$

$$\min_{S\in\theta}\{f_2(S) |\ s.t.\ f_1(S) \le \beta\}$$

Non-Regular Performance Measures

Problems with combinations of criteria are referred to as having **non-regular performance measures**

These problems are often much harder than those with regular performance measures

Example: Minimize total earliness and tardiness:

$$\sum_{j}^{n} E_j + \sum_{j}^{n} T_j$$

is strongly NP-Hard.

Hierarchical Objectives

<u>Another Formulation:</u> In some applications a hierarchy of objectives can be defined

<u>Example:</u> For $1 \mid \mid \sum_j C_j, L_{max}$ the **primary goal** is total completion time and the **secondary goal** is maximum lateness

If $\mathrm{C_{min}}$ is the minimum total completion time and $\theta = \{S \mid \sum_j C_j = C_{min}\}$ then the problem is:

$$\min_{S \in \theta} L_{max}(S)$$

Primary and Secondary Objectives

For $1 \mid \mid \sum_j C_j , L_{max}$:

- If there are no jobs with identical processing time then exactly one schedule minimizes $\sum C_j$

- If there are jobs with identical processing times then select the one that minimizes L_{max}

$\Longrightarrow$ Optimal rule is SPT/EDD

Primary and Secondary Objectives

Example:

Solve the following instance of $1 \mid \mid \sum C_j, L_{max}$

Jobs	1	2	3	4	5	6	7	8	9	10
p_j	5	8	8	3	6	5	4	3	2	3
d_j	5	7	12	3	8	7	5	1	2	4

Primary and Secondary Objectives

Consider the **reverse** case: $1 \mid \mid L_{max}, \sum_j C_j$:

- EDD minimizes L_{max}
- Given $L_{max} = z$ an equivalent problem can be generated by setting **deadlines** $\mathrm{d_j} + z$
- New problem is to minimize $\sum C_j$ subject to **deadline constraints**

IP Formulation: $\sum C_j$ with deadlines

Decision Variables:

$$C_j = completion\ time\ for\ job\ in\ position\ j$$

$$x_{ij} = \begin{cases} 1 & job\ i\ is\ in\ position\ j \\ 0 & otherwise \end{cases}$$

IP Formulation:

$$\min \sum_{j=1}^{n} C_j$$

$$s.t.$$

Assignment constraints $\longrightarrow$

$$\sum_j x_{ij} = 1 \qquad i = 1, \dots, m$$

$$\sum_i x_{ij} = 1, \qquad j = 1, \dots, n$$

Completion time for jobs in position 1 $\longrightarrow$

$$C_1 = \sum_i p_i x_{i1}$$

$$C_j = C_{j-1} + \sum_i p_i x_{ij} \qquad j = 2, \dots, n$$

Deadline constraints $\longrightarrow$

$$C_j \leq \sum_i d_i x_{ij} + z \qquad j = 1, \dots, n$$

$$x_{ij}\ binary, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

Definition

A schedule is called **Pareto-optimal** for a minimization problem if it is not possible to decrease the value of one objective without increasing the value of another.

Weighted Objectives

Another approach to multi-criteria problems is to associate a **weight for each objective**

Given two objectives, γ_1 and γ_2 a schedule with a weighted combination of objectives is:

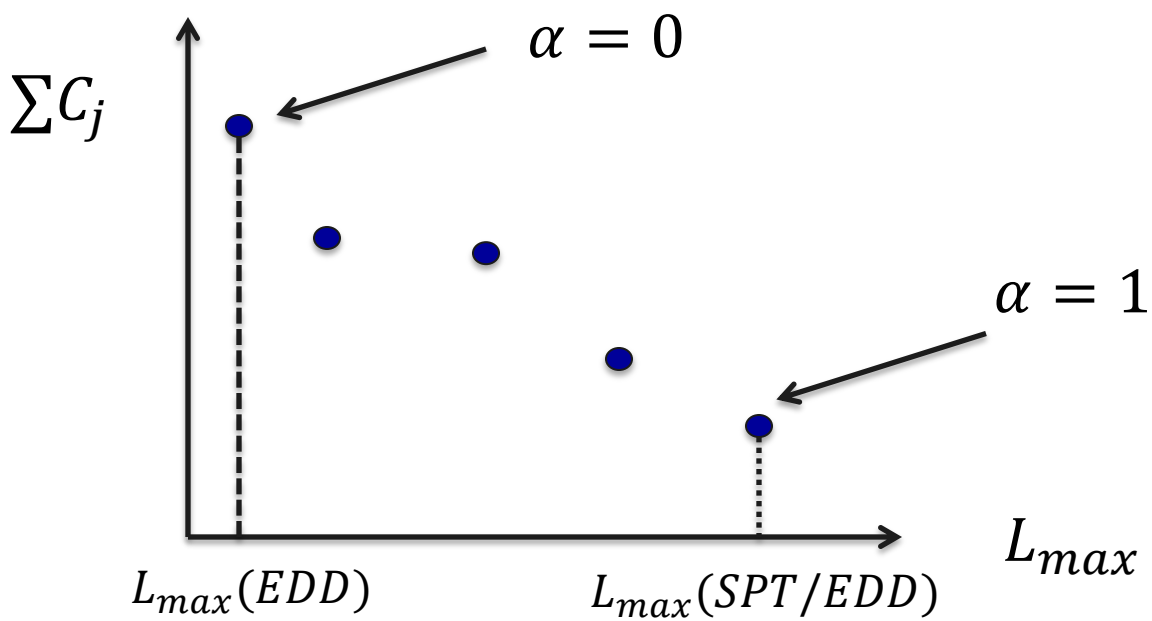
$$a\ |b\ |\alpha\gamma_1 + (1-\alpha)\gamma_2$$

All **Pareto optimal schedules** can be represented by a set of points (γ_1, γ_2), and can be generated by varying α in the range 0 to 1.

Weighted Objectives

Tradeoff between total completion time and maximum lateness:

$$\alpha \sum C_j + (1-\alpha) L_{max}$$



Example: Outpatient Surgery

Patient Intake: administrative activities, pre-surgery exam, gowning, site prep, anesthetic

Surgery: incision, one or multiple procedures, pathology, closing

Recovery: post anesthesia care unit (PACU), ICU, hospital bed

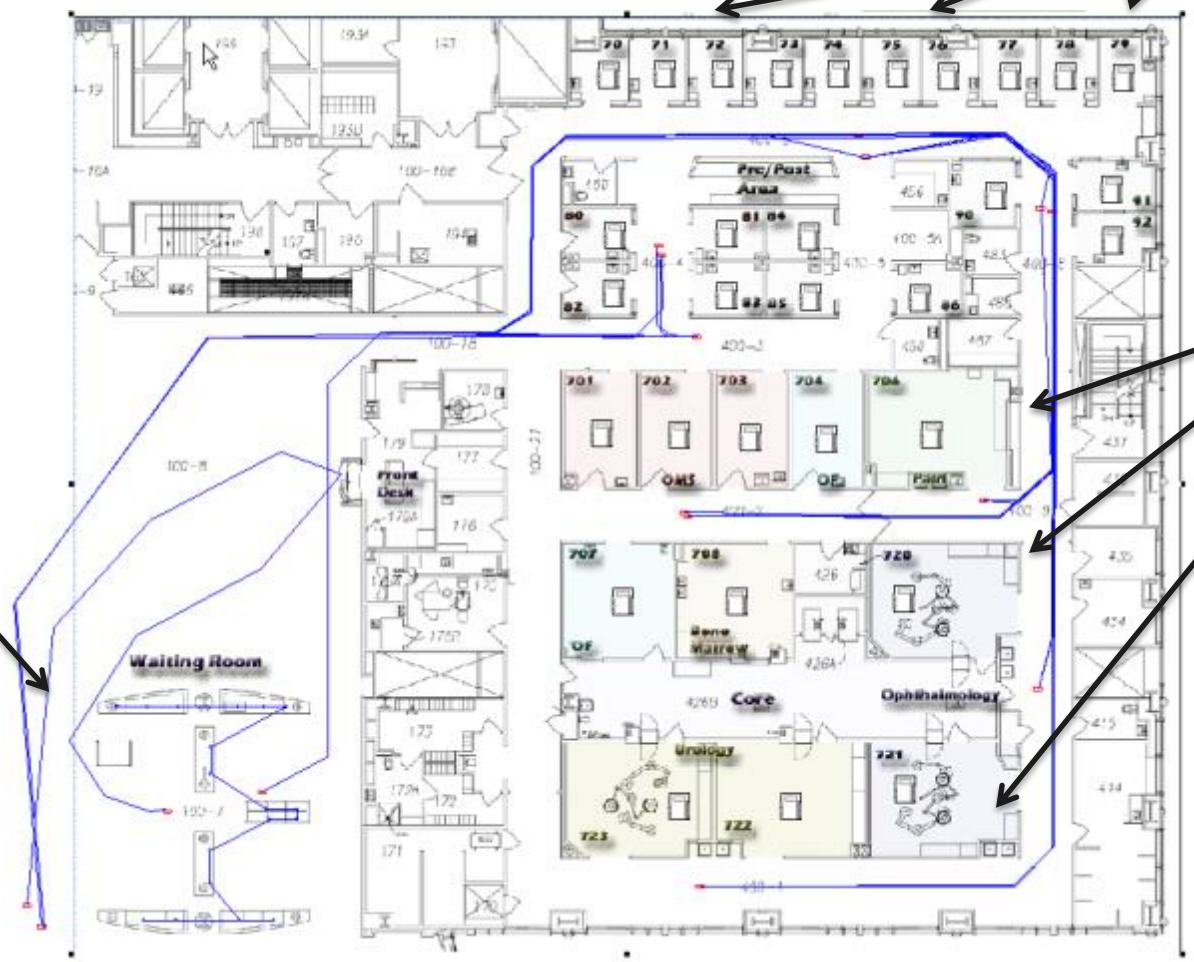
Outpatient Surgery Center

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

Pre/post rooms

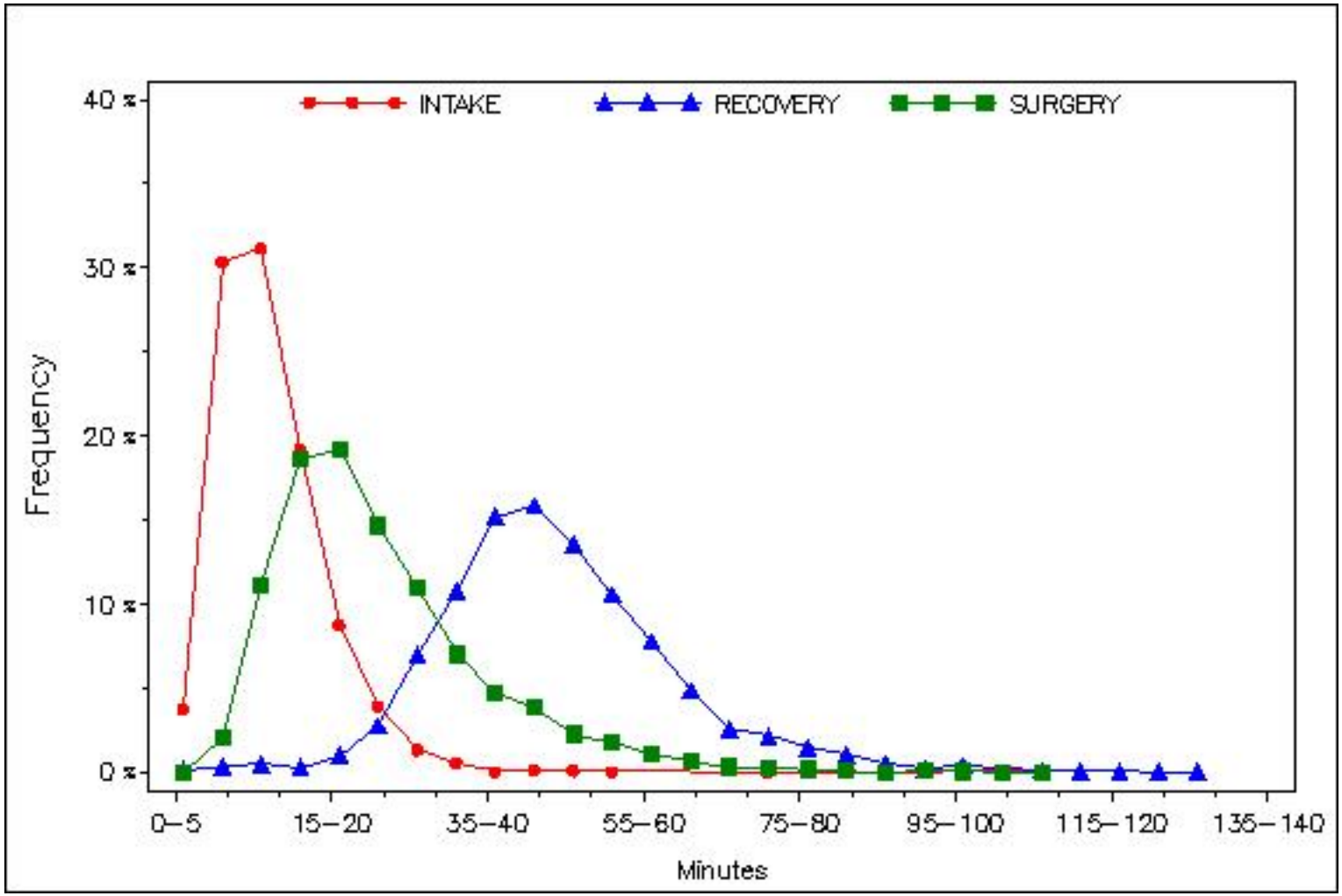
Operating rooms

Patient waiting area

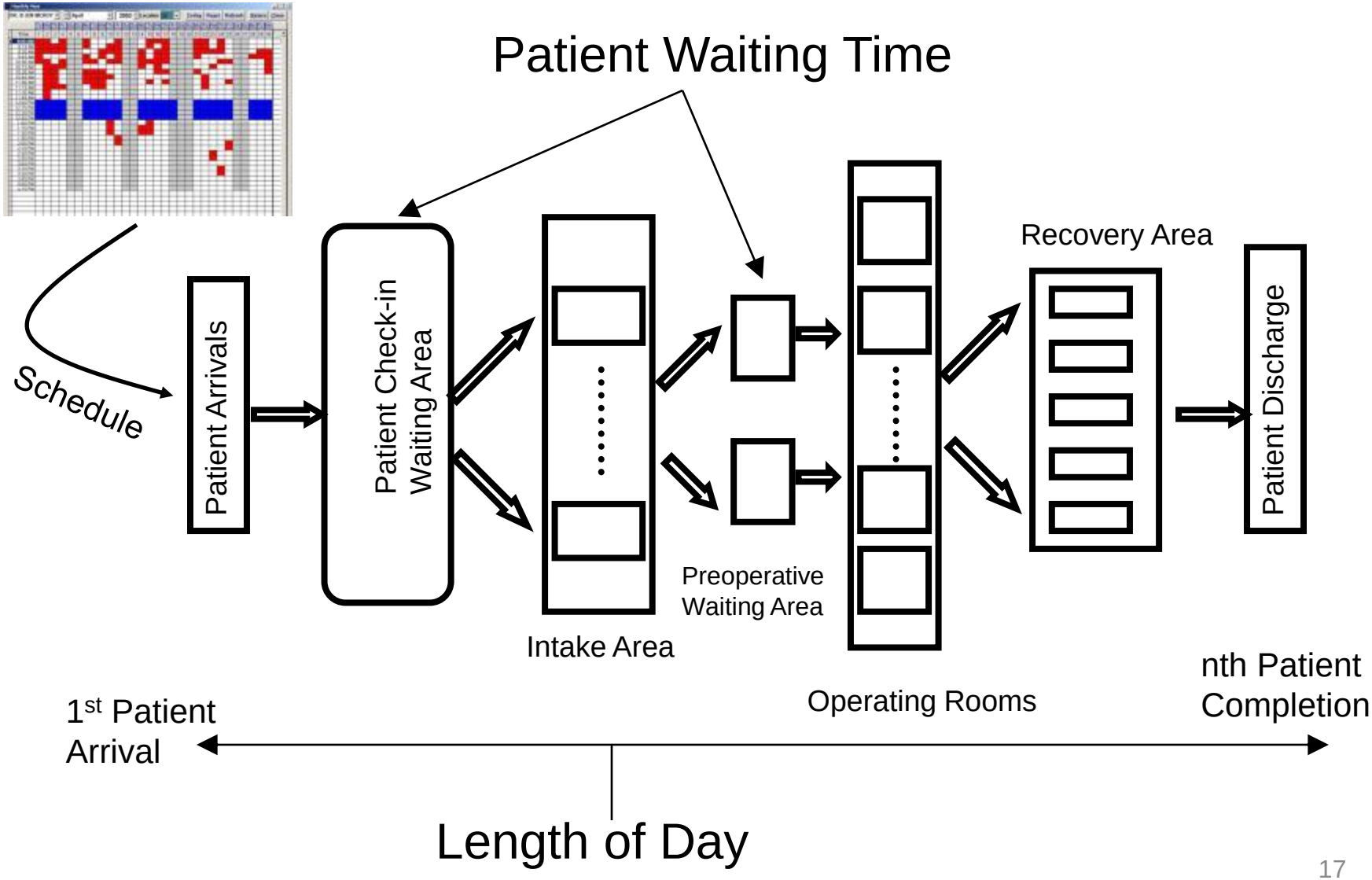


Blue lines represent patient flow

Example: Outpatient Surgery

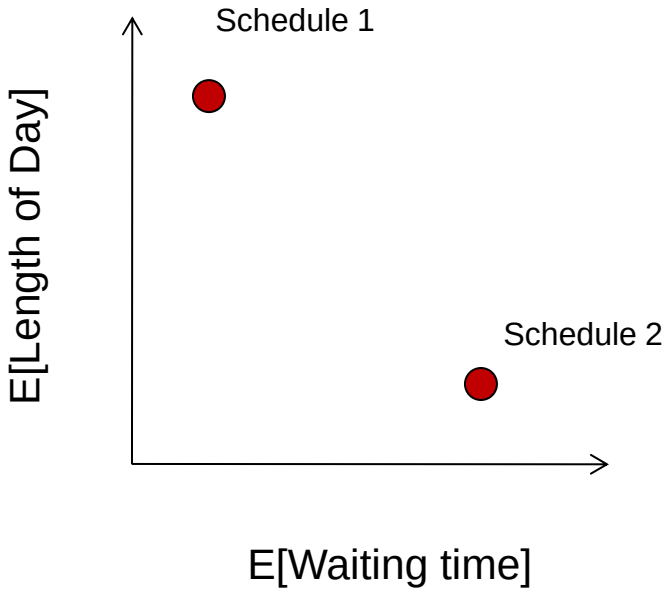


Example: Competing Objectives



Competing Criteria

- Patient Waiting Time:
  - negatively correlated with patient inter-arrival times
  - waiting early in the day causes waiting later in the day
- Length of Day:
  - positively correlated with patient inter-arrival times



Genetic Algorithm

Decision variables: scheduled start times to be assigned to *n* patients each day

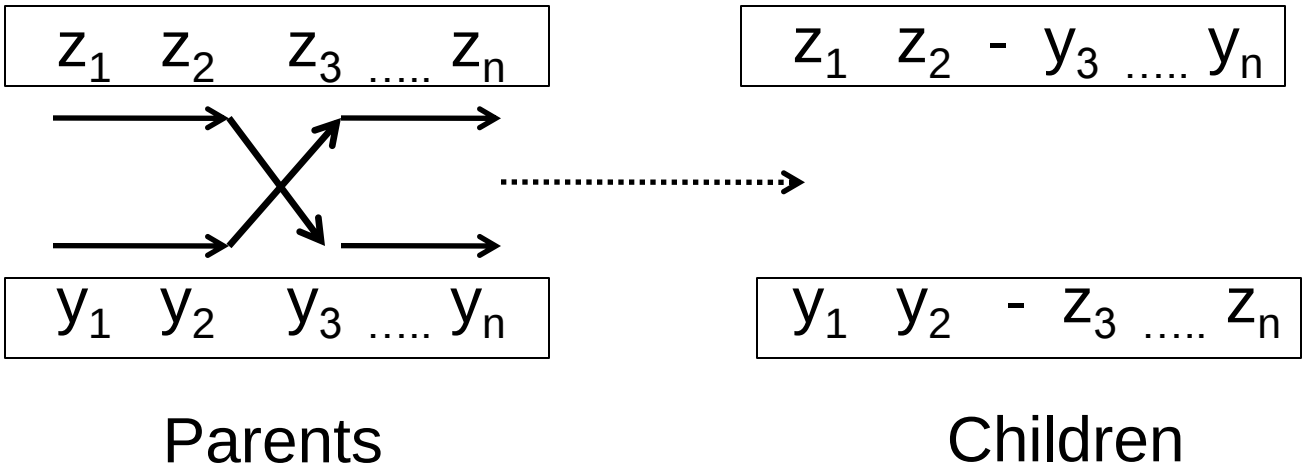
Goal: Generate the set of Pareto optimal schedules to understand tradeoffs between waiting and length of day

- Schedules generated using a genetic algorithm (GA)
- Non-dominated sorting used to identify the Pareto set and feedback into GA

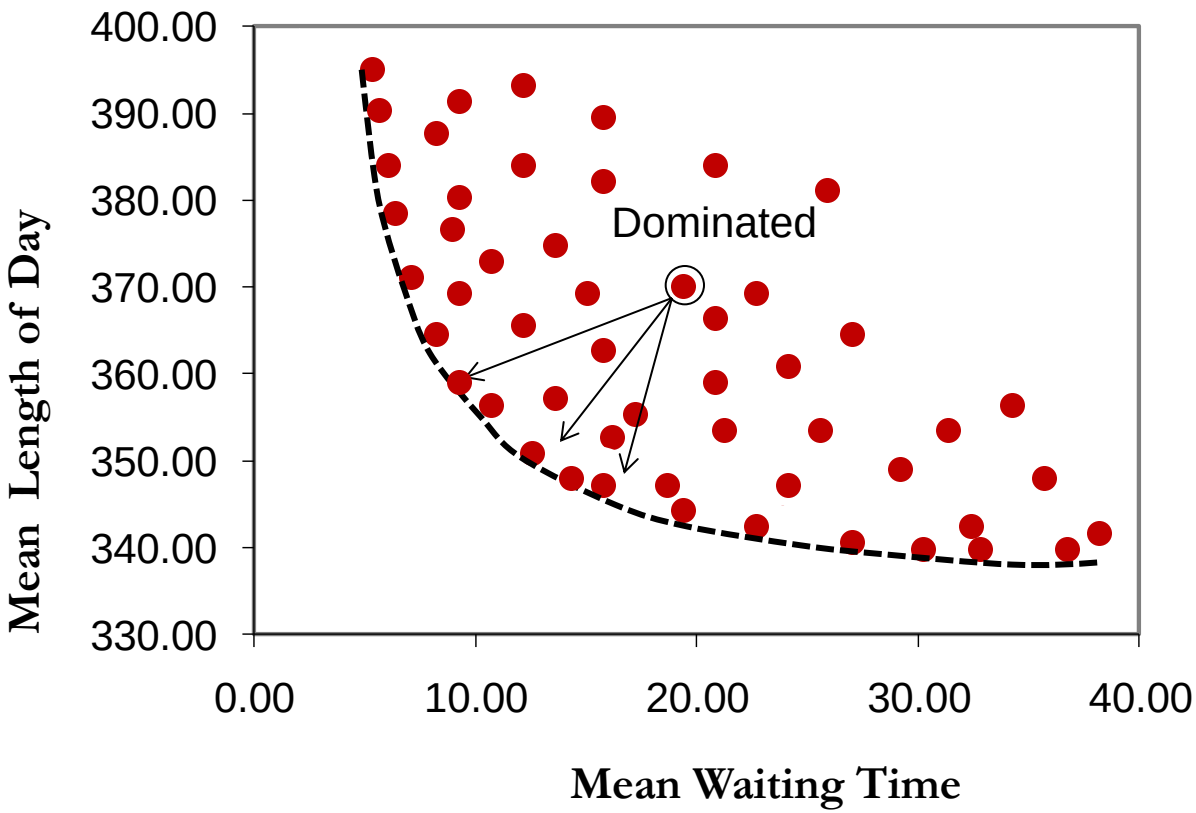
Genetic Algorithm

Main features of the GA:

- Randomly generated initial population of schedules
- Selection based on non-dominated sorting
- Single point crossover:



Pareto Optimal Schedules



New Topic!

- Parallel Machine Scheduling

Parallel Machine Models

- Minimize Makespan
  - Without Preemptions
  - With Preemptions

Minimizing Makespan

- Minimizing makespan without preemptions is NP-Hard since $P2 \mid \mid C_{max}$ is equivalent to PARTITION

Example:

Jobs	1	2	3	4	5	6
p_j	5	4	3	7	1	2

$Pm||C_{max}$

$Pm||C_{max}$ can be formulated as an integer program

$$x_{ij} = \begin{cases} 1 \text{ if job j is assigned to machine i} \\ 0 \text{ otherwise} \end{cases}$$

C_i = completion time of jobs on machine i

$$\min C_{max}$$

$$s.t.$$

$$C_{max} \geq C_i, \qquad i = 1, \dots, m$$

$$\sum_i x_{ij} = 1, \qquad j = 1, \dots, n$$

$$C_i = \sum_j p_j x_{ij}, \qquad i = 1, \dots, n$$

$$x_{ij} \; binary, \quad i = 1, \dots, m, j = 1, \dots, n$$

$Pm \mid\mid C_{max}$

- Because $Pm \mid\mid C_{max}$ is difficult to solve, many **heuristics** have been developed

- The most well known heuristic for $Pm \mid\mid C_{max}$ is the Longest Processing Time First (LPT) heuristic

<u>LPT Heuristic:</u>

Step 1: Order jobs from longest to shortest

Step 2: Allocate the longest unscheduled job to the machine with the earliest completion time

Step 3: If all jobs are scheduled STOP; else return to Step 2

$Pm \mid\mid C_{max}$

- Apply the LPT heuristic to the following instance of $P2 \mid\mid C_{max}$

Example:

Jobs	1	2	3	4	5	6
p_j	5	4	3	7	1	2

LPT Rule for $Pm||C_{max}$

- LPT has the following worst case performance guarantee

<u>Theorem 5.1.1 (Pinedo):</u> For $Pm||C_{max}$

$$\frac{C_{max}(LPT)}{C_{max}(OPT)} \leq \frac{4}{3} - \frac{1}{3m}$$

Proof: Completed in class

<u>Note:</u> The above bound is for the worst possible problem instance. Often the average case performance is much better