Greedy Interval Scheduling

- Interval Scheduling - Generic Greedy Algorithm
- Sub-Optimal Greedy Algorithms
  - Earliest Starting Request First
  - Latest Finishing Request First
  - Shortest Duration Request First
  - Least Conflicting Request First
- Earliest finish time First Algorithm
  - Correctness and Optimality
  - Implementation and Runtime

Imdad ullah Khan

Interval Scheduling: Introduction

- You have a mono-task resource

  ▷ e.g. a lecture room or a research equipment

- and multiple requests to use the resource

  Each request specifies a start time and finish time

- Problem is to schedule (accept/reject) the requests

  Selected requests must not overlap in time

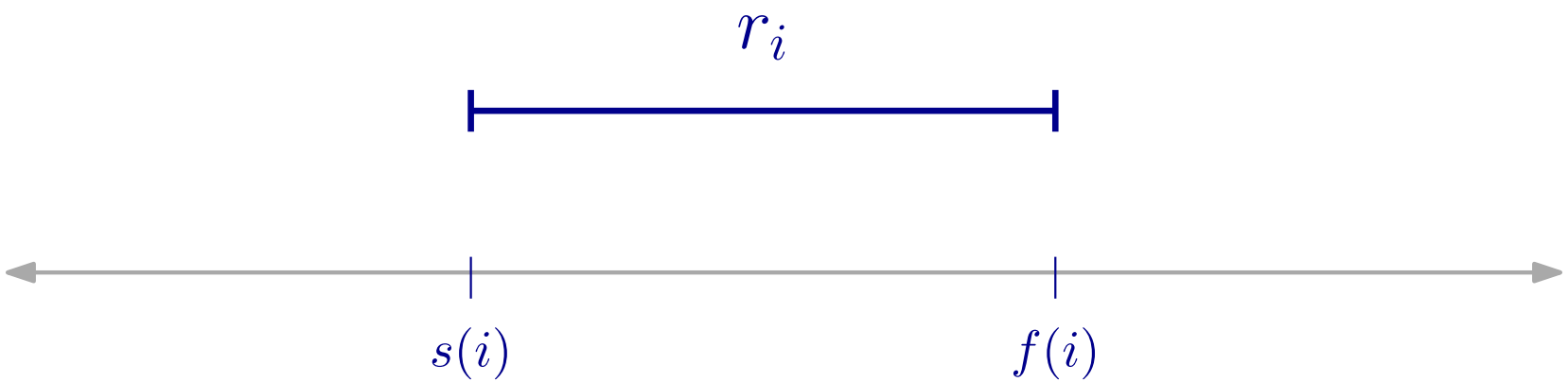
- The goal is to accept the maximum number of requests

Interval Scheduling: Introduction

- $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ (set of requests)
- Starting and finishing time of r_i: $s(i)$ and $f(i)$

$$\text{for } 1 \leq i \leq n \quad s(i) < f(i)$$

- Duration of request r_i is $d_i = f(i)$ - $s(i)$

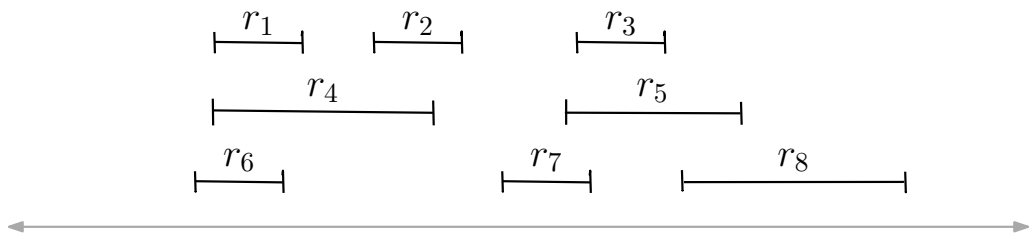


Interval Scheduling: Compatible Requests

Requests r_i and r_j are **compatible** if they do not overlap in time

▷ Otherwise r_i and r_j are **conflicting**

$\underbrace{s(i) < f(i) \quad < \quad s(j) < f(j)}_{r_i \text{ is to the left of } r_j}$ OR $\underbrace{s(j) < f(j) \quad < \quad s(i) < f(i)}_{r_i \text{ is to the right of } r_j}$



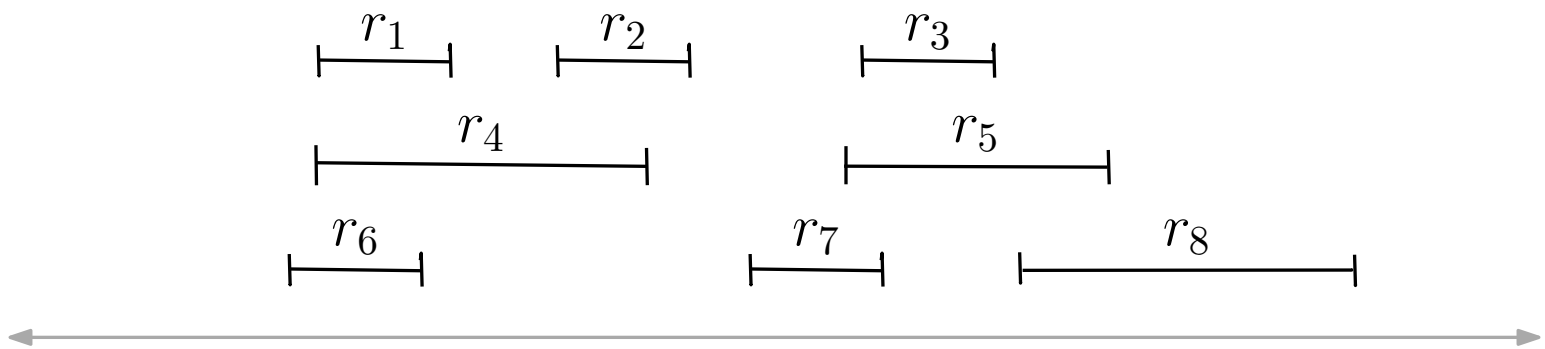
- r_1 and r_2 are compatible, r_4 and r_8 are compatible
- r_1 and r_4 are conflicting, r_5 and r_7 are conflicting

A set is compatible if all pairs in it are compatible

- $\{r_1, r_2, r_8\}$ is compatible
- $\{r_1, r_2, r_5, r_8\}$ is not compatible

Interval Scheduling: Problem Formulation

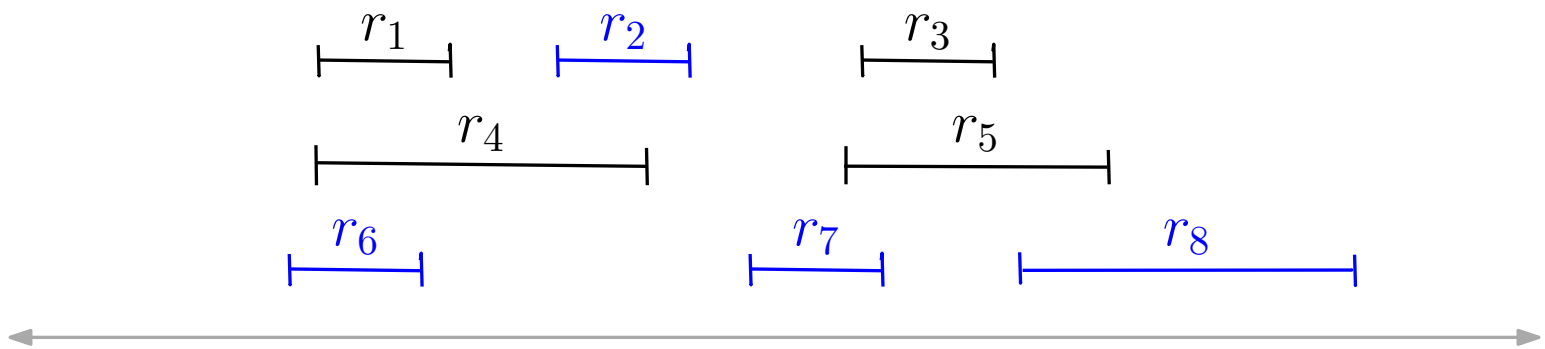
Input: A set $\mathcal{R}$ of requests
Output: A largest compatible subset $S \subset \mathcal{R}$



Interval Scheduling: Problem Formulation

Input: A set $\mathcal{R}$ of requests

Output: A largest compatible subset $S \subset \mathcal{R}$



Interval Scheduling: Greedy Algorithm

Process requests in a fixed order and select compatible requests greedily

Algorithm : Interval Scheduling Algorithm ($\mathcal{R}$)

$A \leftarrow \emptyset$
while $\mathcal{R} \neq \emptyset$ **do**
  select a request r_x from $\mathcal{R}$
  remove from $\mathcal{R}$ all those requests conflicting with r_x
  $A \leftarrow A \cup \{r_x\}$
return A

- By construction the algorithm is correct

▷ (A is a compatible subset)