Scheduling Anomalies

Deterministic Scheduling

- Assume times of tasks (and communication) are known, which they often aren't.
- Problem is NP-hard for all but the most trivial classes of assumptions.
- Unexpected scheduling anomalies.

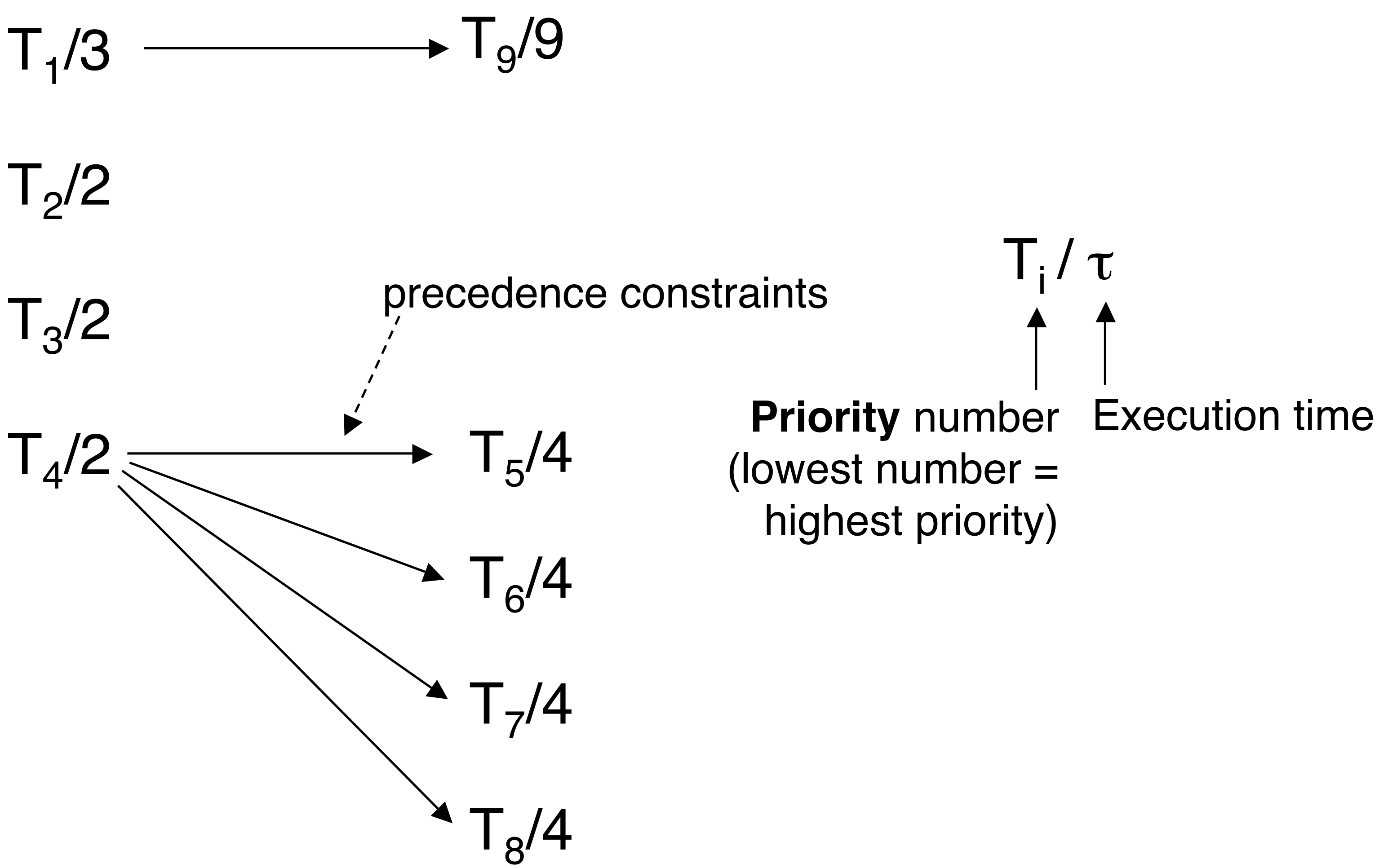
Scheduling Anomalies (R.L. Graham, 1960's)

- The following are expected to reduce overall execution time:
  - Reducing execution times of individual tasks
  - Relaxing precedence constraints between tasks
  - Adding more processors

Scheduling Anomalies (R.L. Graham, 1960's)

- The following are expected to reduce overall execution time:
  - Reducing execution times of individual tasks
  - Relaxing precedence constraints between tasks
  - Adding more processors
- For some algorithms, these can actually *increase* the execution time.

Priority Scheduling Anomalies

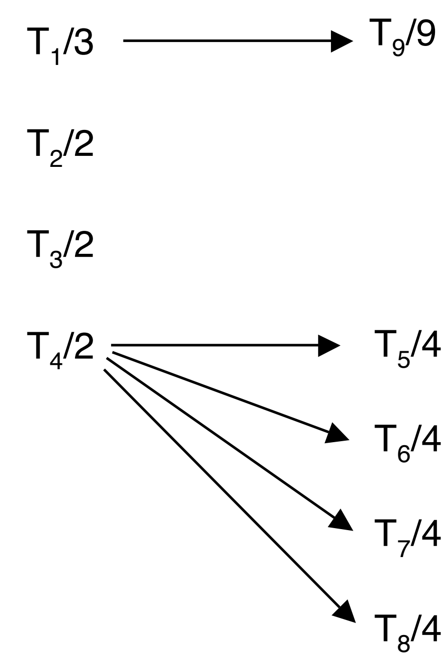


Consider Scheduling on 3 processors

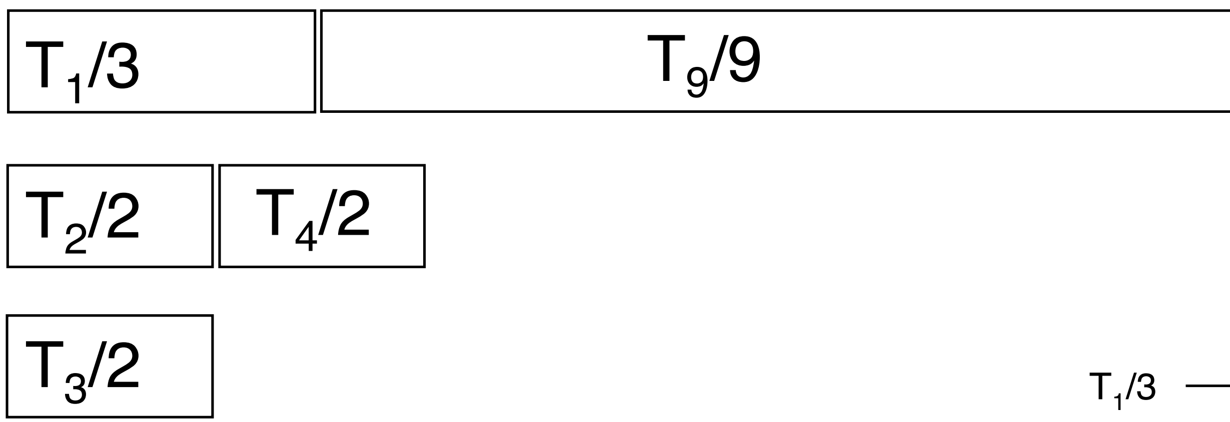
$T_1/3$

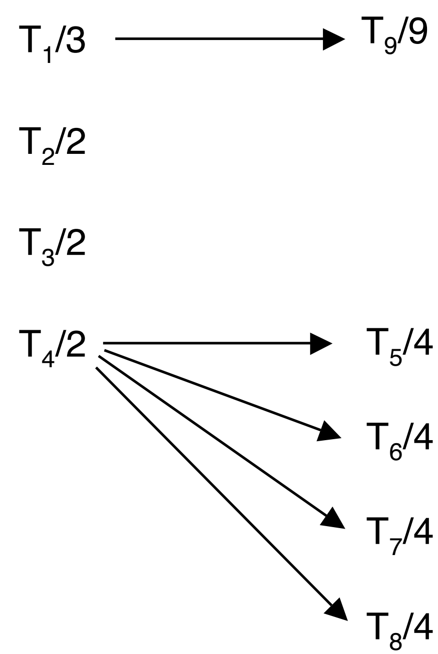
$T_2/2$

$T_3/2$

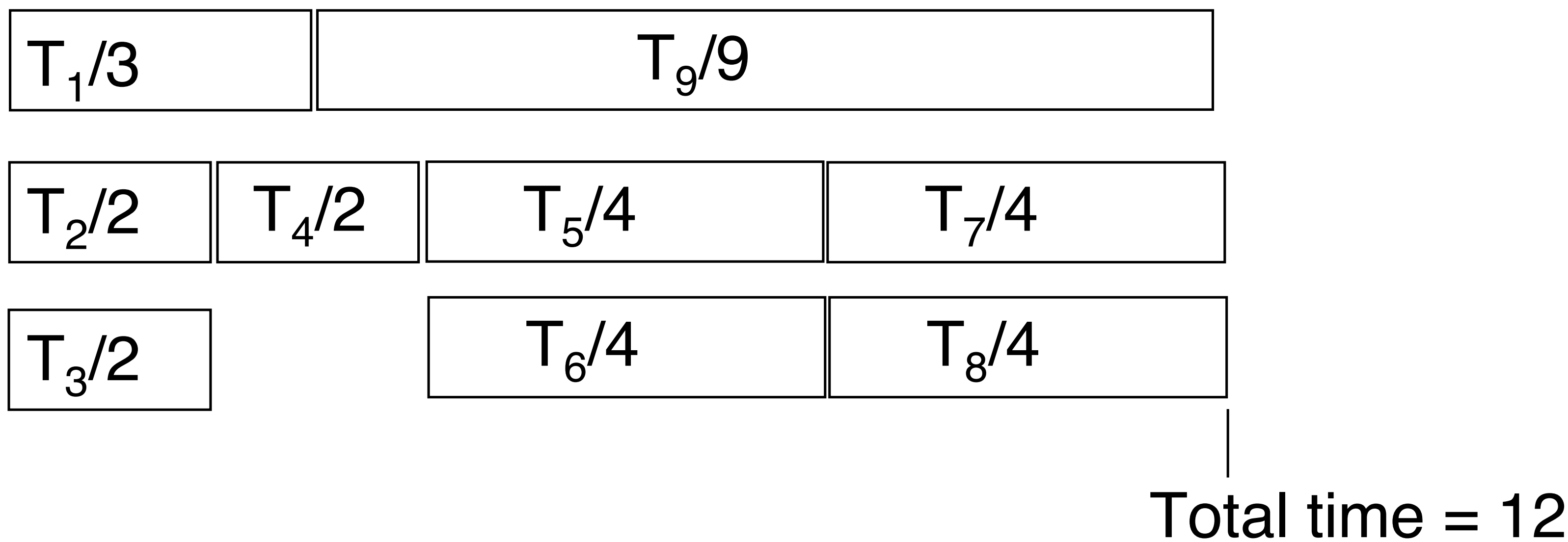


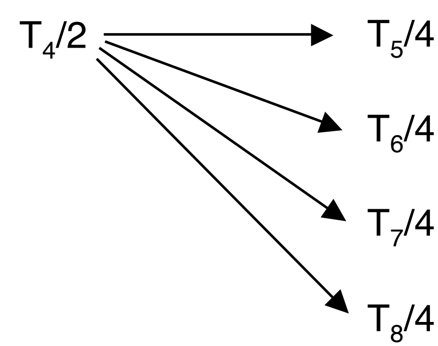
Consider Scheduling on 3 processors





Consider Scheduling on 3 processors





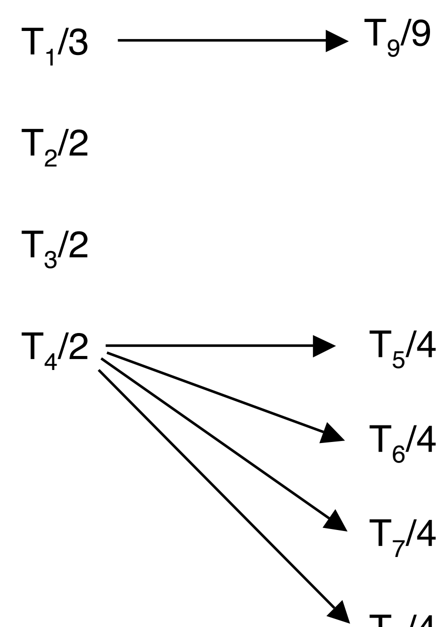
Consider Scheduling on 4 processors

$T_1/3$

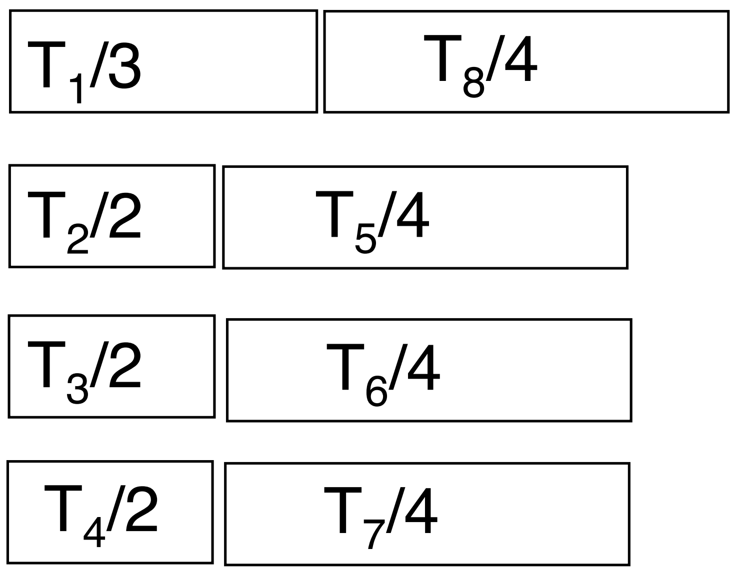
$T_2/2$

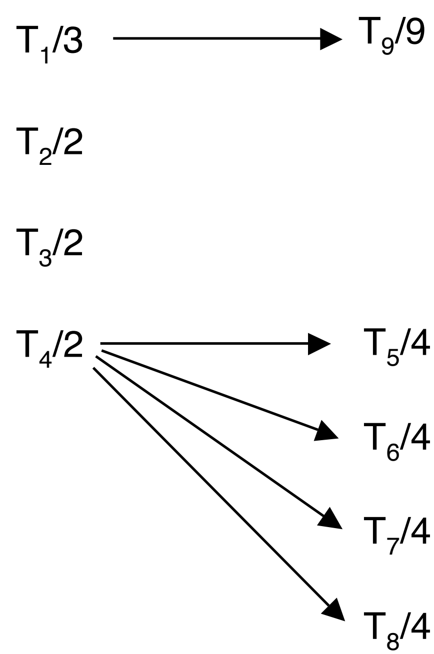
$T_3/2$

$T_4/2$

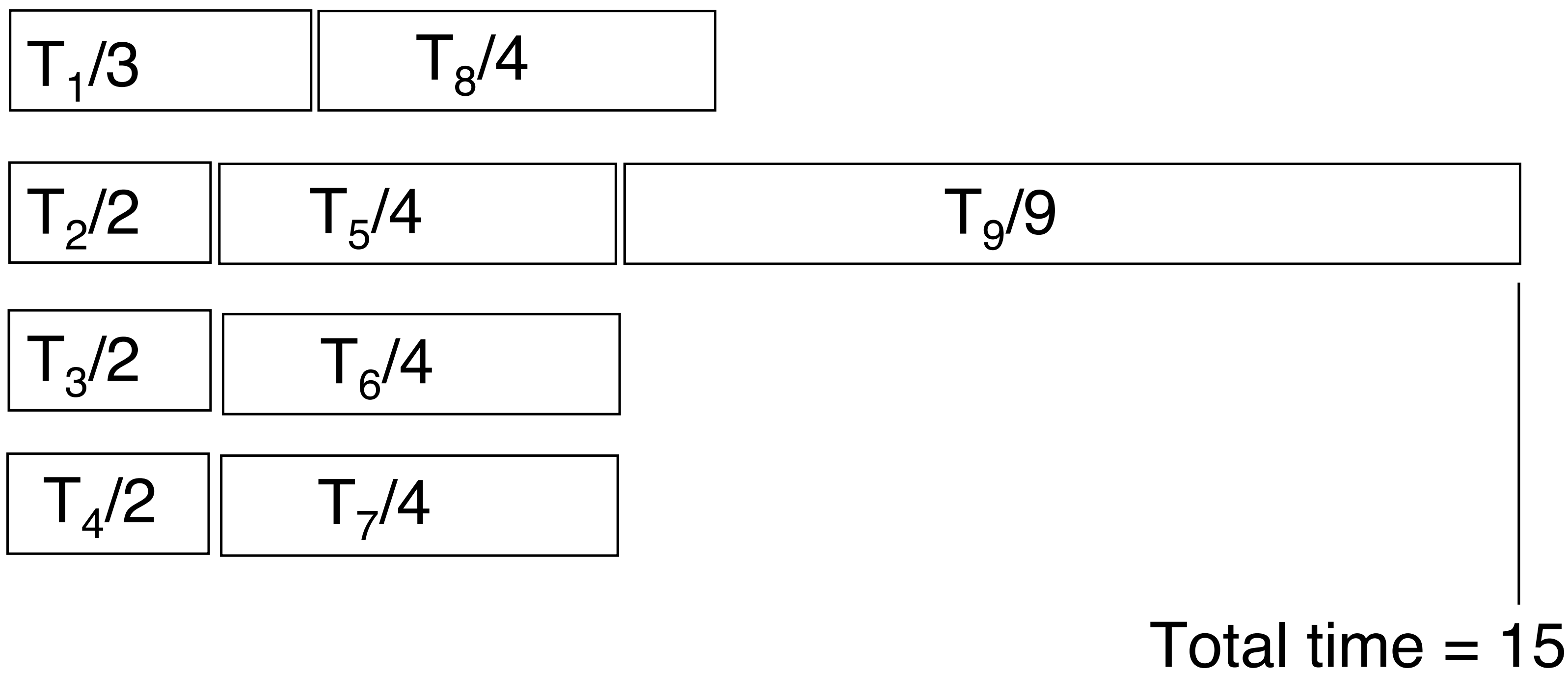


Consider Scheduling on 4 processors

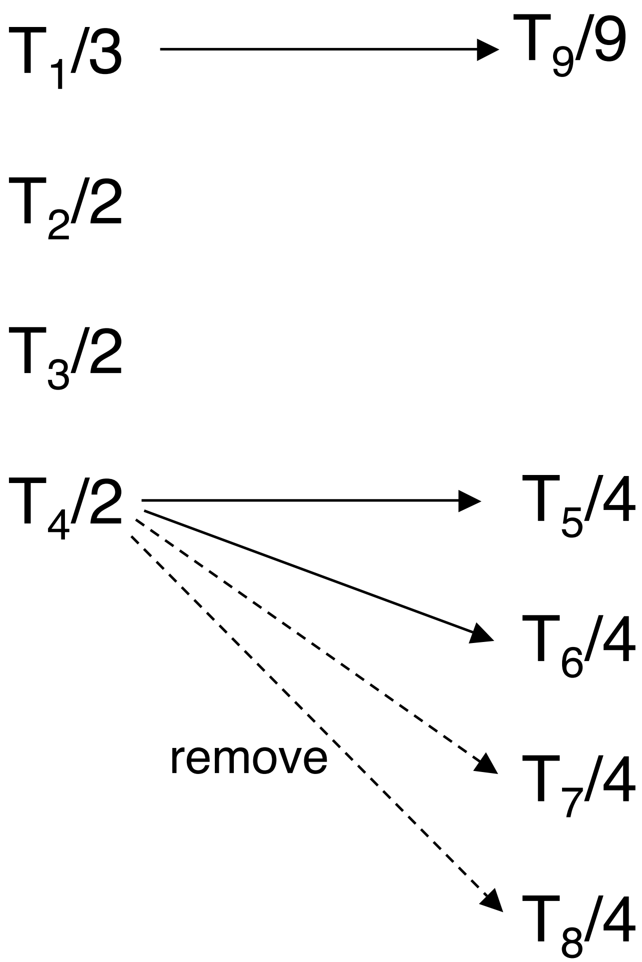




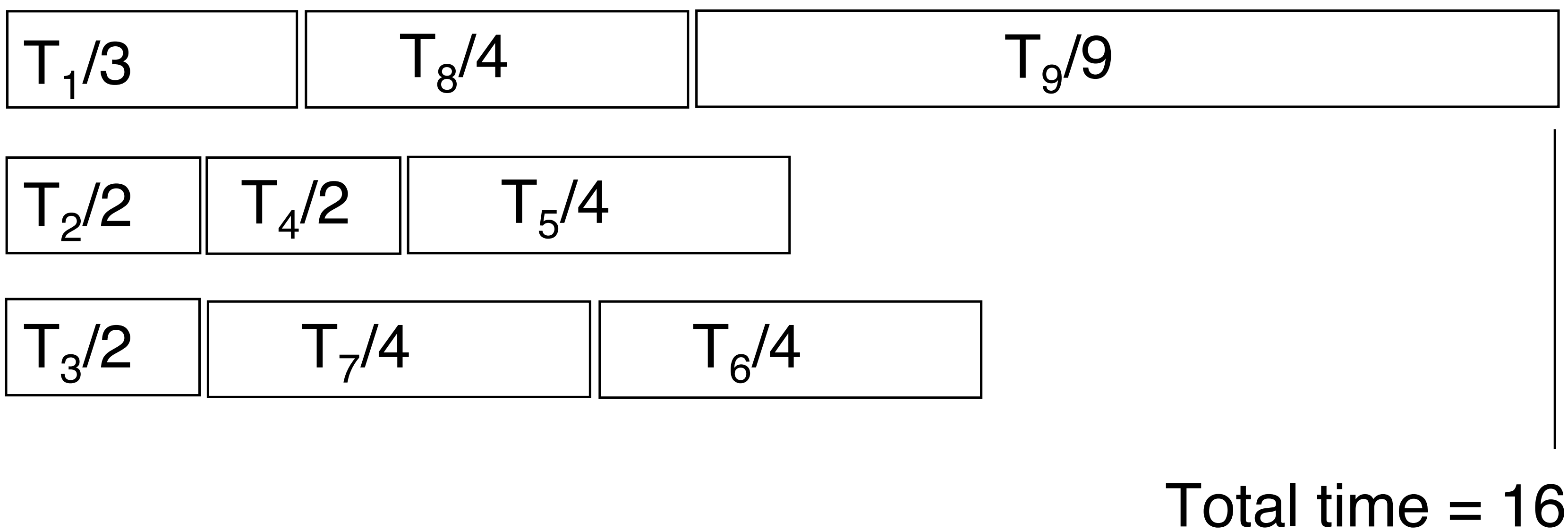
Consider Scheduling on 4 processors



Consider Relaxing Constraints



Consider the relaxed constraints on 3 processors



Cause of Anomalies

- Obviously the anomalies are caused by the use of the **priority rule** in scheduling:
  - This rule is cheap to implement (O(n)).
  - It does not take into account optimizations that would be possible by violating strict priority.

- In general, finding true optimum would entail a search, which tends to be much more expensive.

Bounds on Anomalies
(due to R.L. Graham)

- Let ‘ designate times for system with relaxed constraints and shorter individual times. Then
- Time(m)/ Time(m’) $\leq$ 1 + (m-1)/m’,
  where m’ $\geq$ m are numbers of processors.

- Example: Time(2)/Time(3) $\leq$ 4/3.
- Worst case: Time(m)/Time(m’) < 2.