Quantifying the Exact Sub-optimality of Non-preemptive Scheduling

Robert I. Davis[1], Abhilash Thekkilakattil[2] ,
Oliver Gettings[1], Radu Dobrin[2], Sasikumar Punnekkat[2]

[1]Real-Time Systems Research Group, University of York, UK
[2]Mälardalen Real-Time Research Center, Mälardalen University, Sweden.

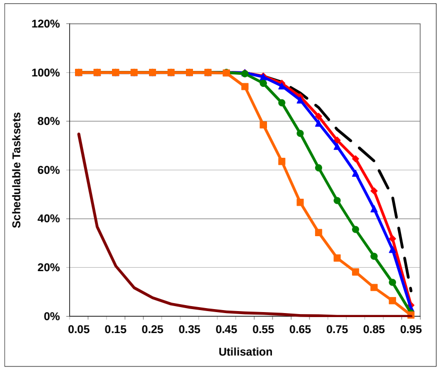
Outline

- **Intro**
  - How do we compare scheduling algorithms
  - Speedup factors and sub-optimality
  - Previous results in this area
- **Exact Speedup factors**
  - EDF-NP v EDF-P
  - FP-NP v EDF-P
  - FP-NP v FP-P
- **Reverse case**
  - FP-P v FP-NP
- **Summary and open problems**

Comparison of scheduling algorithms

- **Empirical methods**
  - Generate lots of task sets
  - Success ratio plots
  - Weighted schedulability graphs – explore performance w.r.t. certain parameters

  Give an average case comparison



- **Theoretical methods**
  - Prove resource augmentation bounds or **speedup factors**

  Give a worst-case comparison

  Focus of this talk

$1/\Omega$

Speedup factors and sub-optimality

Speedup factor (of scheduling algorithm A versus scheduling algorithm B) is the factor by which the speed of the processor needs to be increased, to ensure that any task set that is feasible under algorithm B is guaranteed to be feasible under algorithm A

Sub-optimality: where B is an optimal algorithm, then the speedup factor provides a measure of the sub-optimality of algorithm A

[Note by **feasible**, for fixed priority scheduling, we mean there is some priority asignment with which the task set is schedulable]

Finding exact speedup factors

- **Lower bound on speedup factor**
  - Find a task set that is schedulable under algorithm B and is not schedulable under algorithm A unless the processor speed is increased by at least a factor of X

  X is a lower bound on the speedup factor
- **Upper bound on speedup factor**
  - Prove that any task set that is schedulable under algorithm B is also schedulable under algorithm A on a processor whose speed has been increased by a factor of Y

  Y is an upper bound on the speedup factor
- **Exact speedup factor**
  - When upper and lower bounds are equal

Problem scope

- **Single processor systems**
  - Execution time of all tasks scales linearly with processor clock speed
- **Sporadic task model**
  - Static set of *n* tasks τ_i with priorities 1..*n*
  - Bounded worst-case execution time C_i
  - Sporadic/periodic arrivals: minimum inter-arrival time T_i
  - Relative deadline D_i
  - Independent execution (no resource sharing)
  - Independent arrivals (unknown a priori)

Interested in comparing pre-emptive and non-preemptive scheduling (both EDF and Fixed Priority)

Background: Scheduling algorithms & optimality

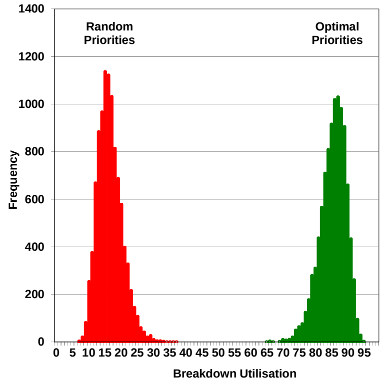
- **Pre-emptive**
  - **EDF-P** is an **optimal** uniprocessor scheduling algorithm for arbitrary-deadline sporadic tasks

  EDF-P dominates FP-P, EDF-NP, and FP-NP

- **Non-pre-emptive**
  - No work-conserving non-preemptive algorithm is optimal
  - Inserted idle time is necessary for optimality
  - **EDF-NP** is **optimal** in a **weak sense** that it can schedule any task set for which a feasible work-conserving non-preemptive schedule exists

  EDF-NP dominates FP-NP

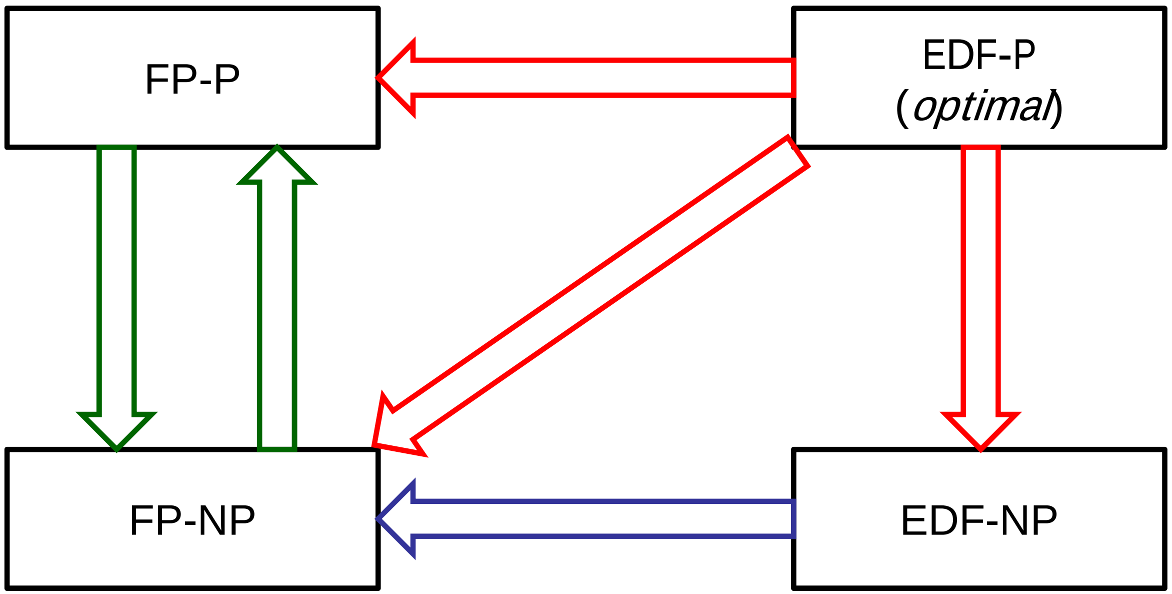
Background:
Scheduling algorithm optimality



- **Fixed Priority Scheduling**
  - Priority assignment important
- **Optimal priority assignment (FP-P)**
  - Implicit-deadlines – Rate-Monotonic
  - Constrained-deadlines – Deadline Monotonic
  - Arbitrary-deadlines – Audsley's Optimal Priority Assignment algorithm
- **Optimal priority assignment (FP-NP)**
  - All 3 cases – Audsley's algorithm

Landscape of scheduling algorithms and speedup factors

Interested in comparing EDF and Fixed Priority (FP) scheduling preemptive and non-preemptive cases



Previous results: Speedup factors for FP-P v. EDF-P and FP-NP v. EDF-NP

As of Jan 2015

Taskset Constraints [Priority ordering]	FP-P v. EDF-P Speedup factor Lower bound	FP-P v. EDF-P Speedup factor Upper bound	FP-NP v. EDF-NP Speedup factor Lower bound	FP-NP v. EDF-NP Speedup factor Upper bound
Implicit-deadline [RM] [OPA]	$1/\ln(2) \approx 1.44269$		$1/\Omega \approx 1.76322$	2
Constrained-deadline [DM] [OPA]	$1/\Omega \approx 1.76322$		$1/\Omega \approx 1.76322$	2
Arbitrary-deadline [OPA] [OPA]	$1/\Omega \approx 1.76322$	2	$1/\Omega \approx 1.76322$	2

Open Problems

Recent results: Speedup factors for FP-P v. EDF-P and FP-NP v. EDF-NP

ECRTS 2015: [van der Bruggen et al.]

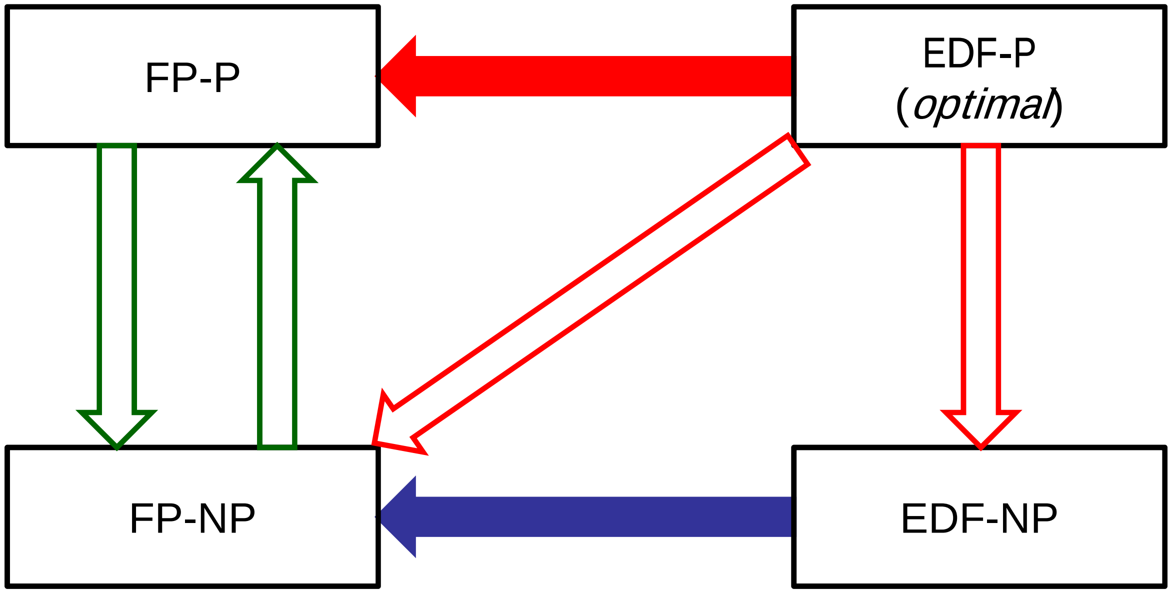
Taskset Constraints [Priority ordering]	FP-P v. EDF-P Speedup factor Lower bound	FP-P v. EDF-P Speedup factor Upper bound	FP-NP v. EDF-NP Speedup factor Lower bound	FP-NP v. EDF-NP Speedup factor Upper bound
Implicit-deadline [RM] [OPA]	**$1/\ln(2) \approx 1.44269$**		**$1/\Omega \approx 1.76322$**	
Constrained-deadline [DM] [OPA]	**$1/\Omega \approx 1.76322$**		**$1/\Omega \approx 1.76322$**	
Arbitrary-deadline [OPA] [OPA]	**$1/\Omega \approx 1.76322$**	**2**	**$1/\Omega \approx 1.76322$**	**2**

Recent results: Speedup factors for FP-P v. EDF-P and FP-NP v. EDF-NP

Real-Time Systems Sept 2015: [Davis et al.]

Taskset Constraints [Priority ordering]	**FP-P v. EDF-P Speedup factor** Lower bound Upper bound	**FP-NP v. EDF-NP Speedup factor** Lower bound Upper bound
Implicit-deadline [RM] [OPA]	**$1/\ln(2) \approx 1.44269$**	**$1/\Omega \approx 1.76322$**
Constrained-deadline [DM] [OPA]	**$1/\Omega \approx 1.76322$**	**$1/\Omega \approx 1.76322$**
Arbitrary-deadline [OPA] [OPA]	**2**	**2**

Focus of this work: Sub-optimality of non-preemptive scheduling



Sub-optimality of EDF-NP and FP-NP

Speedup factors for FP-NP v. FP-P and vice-versa since they are incomparable

Long task problem

- **Non-preemptive scheduling suffers from the long task problem**
  - If $C_{\max} > D_{\min}$ task set is not schedulable
  - Without accounting for this, speedup factor is arbitrarily large

- **Express speedup factor in a way that is parametric in** $C_{\max} / D_{\min}$
  - Simplest form that gives a finite speedup factor

Recap: Schedulability analysis

- **EDF-P Exact test (arbitrary deadlines)**

$$\sum_{\forall \tau_i \in \Gamma} DBF_i(t) \le t$$

$$DBF_i(t) = \max\left(0, \left\lfloor \frac{t - D_i}{T_i} \right\rfloor + 1\right) C_i$$

- **FP-P Exact test (constrained deadlines)**

$$R_i^P = C_i + \sum_{\forall \tau_j \in hp(i)} \left\lceil \frac{R_i^P}{T_j} \right\rceil C_j$$

Recap: Schedulability analysis

- **FP-NP Sufficient test (arbitrary deadlines)**

$$B_i + \sum_{\forall \tau_j \in hep(i)} \left\lceil \frac{D_i}{T_j} \right\rceil C_j \le D_i \quad \textbf{where} \quad B_i = \begin{cases} \max\limits_{\forall \tau_k \in lp(i)} (C_k - \Delta) & i < n \\ 0 & i = n \end{cases}$$

- **FP-NP Sufficient test (constrained deadlines)**

$$W_i^{NP} = C_{\max} + \sum_{\forall \tau_j \in hp(i)} \left\lceil \frac{W_i^{NP} + \Delta}{T_j} \right\rceil C_j$$

$$R_i^{NP} = W_i^{NP} + C_i$$

THE UNIVERSITY of York

RTS York

Exact sub-optimality of EDF-NP

Lower bound on speedup factor for non-preemptive v. preemptive

- **Proof sketch (Lemma IV.3)**
  - Find a task set that requires at least this increase in speed
- **Example task set**

$$\tau_1: C_1 = k-1, D_1 = k, T_1 = k$$

$$\tau_2: C_2 = k^2+1, D_2 = \infty, T_2 = \infty$$

  - Trivially schedulable with preemptive algorithms (EDF-P or FP-P)
  - FP-NP and EDF-NP need to accommodate jobs of both tasks within shorter deadline

$$S \geq (k^2 + k)/k = k+1 \quad \text{since} \quad \frac{C_{\max}}{D_{\min}} = k + \frac{1}{k} \quad \text{then} \quad S \geq 1 + \frac{C_{\max}}{D_{\min}} - \frac{1}{k}$$

- **Lower bound** $S = 1 + \frac{C_{\max}}{D_{\min}}$

Holds for implicit, constrained, or arbitrary deadlines
FP-NP or EDF-NP v. FP-P or EDF-P

Exact sub-optimality of EDF-NP

- **Upper bound**
  - Abugchem et al. [1] (Embedded Systems Letters 2015)

$$S = 1 + \frac{C_{\max}}{D_{\min}}$$

  - Holds for arbitrary deadlines

- **Exact sub-optimality of EDF-NP (speedup factor v. EDF-P)**
  - Upper bound and lower bound are equal (for implicit, constrained, and arbitrary deadlines)

$$S = 1 + \frac{C_{\max}}{D_{\min}}$$

Exact sub-optimality of FP-NP

Upper bound on speedup factor for FP-NP v. EDF-P

- **Proof sketch (Lemma IV.1)**
  - Show speedup factor which is enough for to ensure schedulability under FP-NP using sufficient test and DMPO
- **From definition of** $DBF(t)$

$$\sum_{\forall \tau_j \in \Gamma} DBF_j(2D_i) \geq \sum_{\forall \tau_j : D_j \leq D_i} \left\lceil \frac{D_i}{T_j} \right\rceil C_j \geq \sum_{\forall \tau_j \in hep(i)} \left\lceil \frac{D_i}{T_j} \right\rceil C_j$$

- **FP-NP Sufficient test (arbitrary deadlines)**

$$\frac{C_{\max} + \sum_{\forall \tau_j \in \Gamma} DBF_j(2D_i)}{S} \leq D_i$$

Upper bound on speedup factor for FP-NP v. EDF-P

- **Schedulable under EDF-P on processor of speed 1**

$$\sum_{\forall \tau_j \in \Gamma} DBF_j(2D_i) \le 2D_i$$

Substituting: $\frac{C_{\max} + 2D_i}{S} \le D_i$ assures schedulability under FP-NP

- **Upper bound**

$$S = 2 + \frac{C_{\max}}{D_{\min}}$$

Holds for arbitrary deadlines

Also holds for FP-NP v. FP-P (since EDF-P dominates FP-P)

Lower bound on speedup factor for FP-NP v. FP-P

- **Proof sketch (Lemma IV.3)**
  - Find a task set that requires at least this increase in speed
- **Example task set**

  τ_i: $i = 1..k-1$, $C_1 = 1$, $D_1 = k+1$, $T_1 = k$ (arbitrary deadlines)

  τ_k: $C_k = 1$, $D_k = k+1$, $T_k = k+1$

  τ_{k+1}: $C_{k+1} = k^2$, $D_{k+1} = \infty$, $T_{k+1} = \infty$
- **Schedulability under FP-P**
  - Trivially schedulable on a processor of speed 1
  - Each task τ_j: $j = 1..k$ has a response time of j
  - Task τ_{k+1} executes for 1 unit in the LCM of the higher priority tasks and has a response time of $k^3(k+1)$

Lower bound on speedup factor for FP-NP v. FP-P

- **Schedulability under FP-NP (Lemma IV.5)**
  - Audsley's algorithm for optimal priority assignment
  - Task τ_{k+1} schedulable at the lowest priority (on a processor of speed 1 or higher) so placed at the lowest priority
  - Two distinct cases to consider depending on whether task τ_k or one of the other tasks is assigned the next higher priority
  - Each case has two possibilities to ensure schedulability - see paper
  - Weakest constraint **necessary** for schedulability under FP-NP
    - First jobs of all tasks and second jobs of tasks τ_1 to τ_{k-2} must complete by the deadline at $k+1$ so $S \geq (k^2 + 2k - 2)/(k+1)$
    - As $C_{\max} / D_{\min} = k^2 /(k+1)$

$S \geq \dfrac{2k-2}{k+1} + \dfrac{C_{\max}}{D_{\min}}$ and hence **lower bound is** $S = 2 + \dfrac{C_{\max}}{D_{\min}}$

Also holds for FP-NP v. EDF-P as EDF-P dominates FP-P

Note arbitrary deadlines only

Exact sub-optimality FP-NP v. EDF-P

- **Exact sub-optimality of FP-NP (v. EDF-P)**
  - Upper bound and lower bound are equal (for arbitrary deadlines)

$$S = 2 + \frac{C_{\max}}{D_{\min}}$$

- **Upper and lower bounds on sub-optimality of FP-NP (v. EDF-P)**
  - Implicit and constrained deadlines

Lower bound $S = 1 + \frac{C_{\max}}{D_{\min}}$ Upper bound $S = 2 + \frac{C_{\max}}{D_{\min}}$

Currently an open problem to close the gap and find an exact value

Exact speedup factor for FP-NP v. FP-P

Upper bound speedup factor FP-NP v. FP-P (constrained deadlines)

- **Proof sketch (Lemma IV.4)**
  - Consider any task set that is schedulable on a processor of speed 1 under FP-P with (optimal) DMPO show that it is also schedulable on a processor of speed S under FP-NP with DMPO (not optimal, but suffices to show feasibility)

$$E_i^P(t) = C_i + \sum_{\forall \tau_j \in hp(i)} \left\lceil \frac{t}{T_j} \right\rceil C_j \qquad E_i^P(W_i^P) = W_i^P$$

Response time with FP-P

$$E_i^{NP}(t) = \sum_{\forall \tau_j \in hp(i)} \left\lceil \frac{t+\Delta}{T_j} \right\rceil C_j \qquad E_i^{NP}(W_i^{NP}) + C_{\max} + C_i = W_i^{NP} + C_i$$

Start time with FP-NP

  - Observe

$$E_i^{NP}(t-x) + C_i \le E_i^P(t)$$

$$\forall x \ge \Delta \quad \forall t \ge x$$

Upper bound speedup factor FP-NP v. FP-P (constrained deadlines)

- **Ensure FP-NP schedulability on a processor of speed S**

  Case 1: $W_i^P \geq D_{\min}$

  - Make completion under FP-NP at speed S no later than for FP-P at speed 1, so start time no later than $W_i^P - C_i / S$
  - Sufficient test for FP-NP will give a response time $\leq W_i^P$ if

$$\frac{C_{\max} + E_i^{NP}(W_i^P - C_i / S) + C_i}{S} \leq W_i^P$$

> Blocking + interference before starting + execution

  - Since $E_i^{NP}(W_i^P - C_i / S) + C_i \leq E_i^P(W_i^P) = W_i^P$ substitution gives following condition on schedulability $S \geq 1 + \frac{C_{\max}}{W_i^P}$

Upper bound $S = 1 + \frac{C_{\max}}{D_{\min}}$

Upper bound speedup factor FP-NP v. FP-P (constrained deadlines)

- **Ensure FP-NP schedulability on a processor of speed S**

  Case 2: $W_i^P < D_{\min}$

  - Assume completion under FP-NP at speed S is no later than $D_{\min}$
  - Sufficient test for FP-NP will give a response time $\leq D_{\min}$ if

$$\frac{C_{\max} + E_i^{NP}(D_{\min} - C_i / S) + C_i}{S} \leq D_{\min}$$

  - Since $E_i^{NP}(W_i^P - C_i / S) + C_i \leq E_i^P(W_i^P) = W_i^P < D_{\min}$ substitution gives following condition on schedulability

$$S \geq 1 + \frac{C_{\max}}{D_{\min}}$$

Upper bound $S = 1 + \frac{C_{\max}}{D_{\min}}$

Holds for implicit and constrained deadlines, but not arbitrary deadlines due to schedulability test used in proof

Exact speedup factor FP-NP v. FP-P

- **Arbitrary Deadlines: Lower bound and upper bound are equal => exact speedup factor**

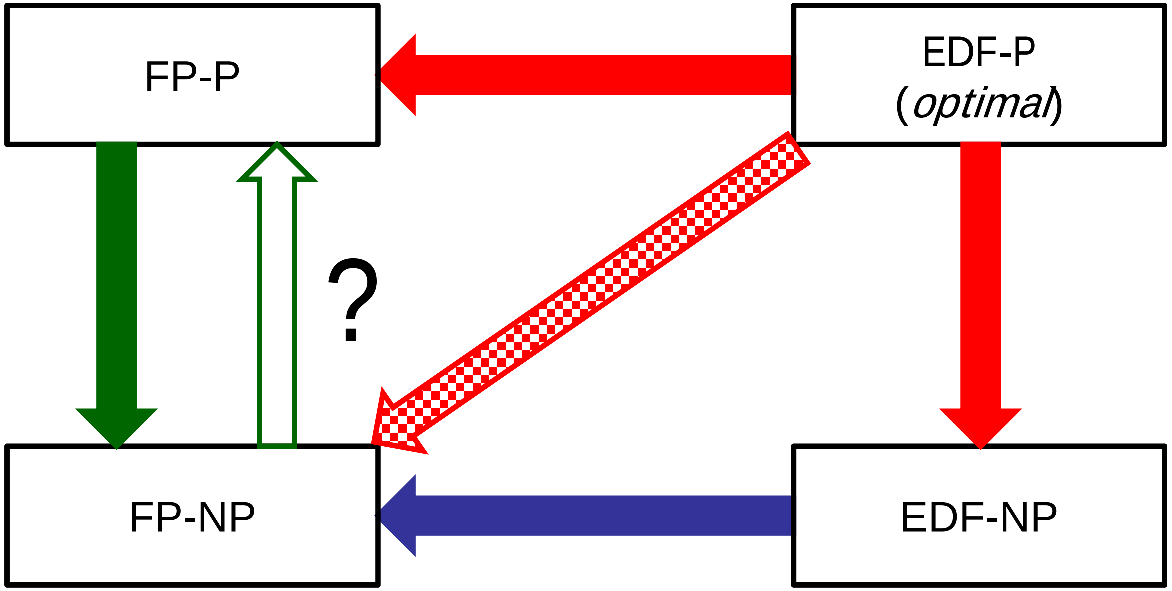
$$S = 2 + \frac{C_{\max}}{D_{\min}}$$

- **Implicit and Constrained Deadlines: Lower bound and upper bound are equal => exact speedup factor**

$$S = 1 + \frac{C_{\max}}{D_{\min}}$$

Interesting that relaxing the task model to arbitrary deadlines adds 1 to the speedup factor needed

Sub-optimality and speedup factors



- Closed speedup factors for FP-NP v. FP-P and EDF-NP v. EDF-P
- Main result for FP-NP v. EDF-P proved (arbitrary deadlines)
  - Remains to close the gap between upper and lower bounds for implicit and constrained deadline cases
- Speedup factor for FP-P v. FP-NP since they are incomparable?

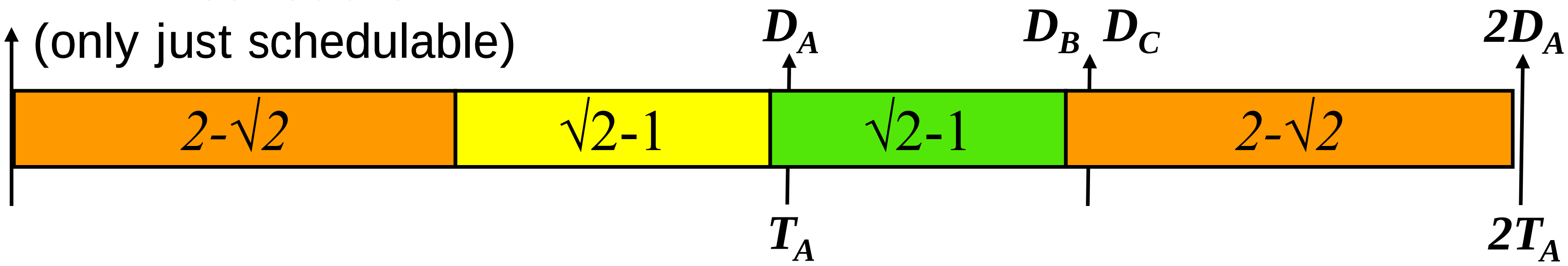
THE UNIVERSITY of York

RTS York

Speedup factor for FP-P v. FP-NP

Lower bounds on speedup factor for FP-P v. FP-NP

FP-NP schedule
(only just schedulable)



- **Task set**

  τ_A: $C_A = \sqrt{2}-1$, $D_A = 1$, $T_A = 1$

  τ_B: $C_B = (2 - \sqrt{2})/2$, $D_B = \sqrt{2}$, $T_B = \infty$

  τ_C: $C_C = (2 - \sqrt{2})/2$, $D_C = \sqrt{2}$, $T_C = \infty$

  Constrained deadlines, DM optimal for FP-P

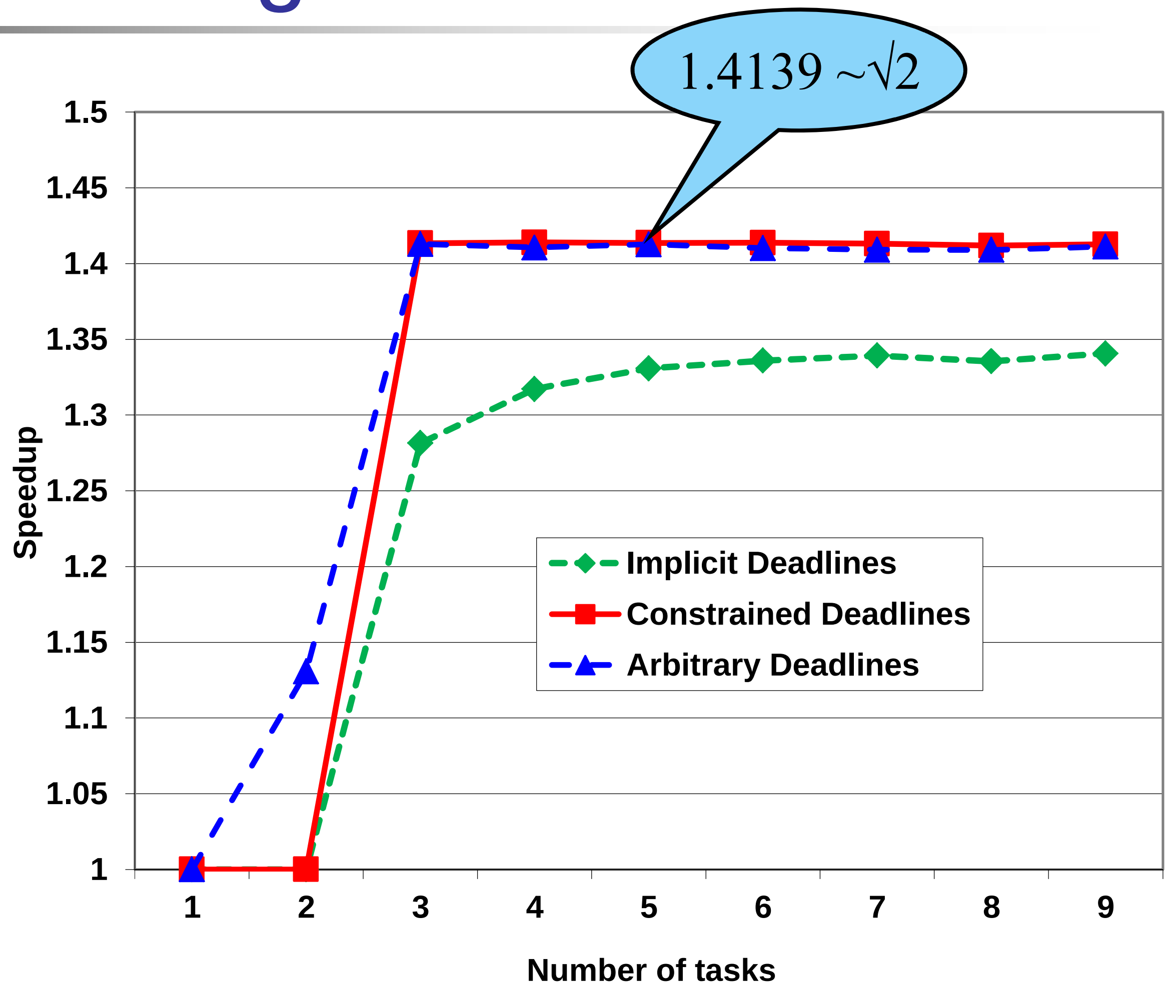
Scale by a factor of $\sqrt{2}$ just schedulable with FP-NP
Lower bound on speedup factor is $\sqrt{2}$

Empirical investigation

Genetic algorithm used to search for task sets requiring a high speedup factor

Highest value found (1.4139)
Very close to √2 for three or more tasks with constrained or arbitrary deadlines

Fairly compelling result since with 3 tasks there are few parameters, so search using GA is very effective



Open problem

- **What is the exact speedup factor for FP-P v. FP-NP?**
  - **Upper bounds** are:
    - **2** for arbitrary deadlines
    - $1/\Omega \approx \mathbf{1.76322}$ for constrained deadlines
    - $1/\ln(2) \approx \mathbf{1.44269}$ for implicit deadlines

    As EDF-P can schedule any task set that is schedulable by FP-NP and those are the speedup factors for FP-P v. EDF-P
  - **Lower bound** is $\sqrt{2}$ for three or more tasks and constrained/arbitrary deadlines
  - Empirically it appears this lower bound may be tight

Proof needed…

Summary: Speedup factors for non-preemptive scheduling

Taskset Constraints [Priority ordering]	FP-NP v. EDF-P Sub-optimality: Lower bound	FP-NP v. EDF-P Sub-optimality: Upper bound	FP-NP v. FP-P Speedup factor	EDF-NP v. EDF-P Sub-optimality
Implicit-deadline [RM] [OPA]	$1+\frac{C_{\max}}{D_{\min}}$	$2+\frac{C_{\max}}{D_{\min}}$	$1+\frac{C_{\max}}{D_{\min}}$	$1+\frac{C_{\max}}{D_{\min}}$
Constrained-deadline [DM] [OPA]	$1+\frac{C_{\max}}{D_{\min}}$	$2+\frac{C_{\max}}{D_{\min}}$	$1+\frac{C_{\max}}{D_{\min}}$	$1+\frac{C_{\max}}{D_{\min}}$
Arbitrary-deadline [OPA] [OPA]	$2+\frac{C_{\max}}{D_{\min}}$		$2+\frac{C_{\max}}{D_{\min}}$	$1+\frac{C_{\max}}{D_{\min}}$

Open Problem

Contribution

Summary:
FP-P v. FP-NP

Taskset Constraints [Priority ordering]	FP-P v. FP-NP Speedup factor Lower bound	Upper bound
Implicit-deadline [RM] [OPA]	**1.34 (expt)**	**1/ln(2) ≈ 1.44269**
Constrained-deadline [DM] [OPA]	$\sqrt{2}$	**1/Ω ≈ 1.76322**
Arbitrary-deadline [OPA] [OPA]		2

Open Problem

Contribution

Questions?