Response-Time Analysis for Mixed Criticality Systems

S.K. Baruah, A. Burns and R.I. Davis

Background

- Mixed criticality systems are becoming a distinct focus for research and industrial application
- Two key issues:
  1. Run-time Robustness
  2. Static Verification
- This paper focuses on the latter

Requirements

- In any multi-application system, failures must be confined to the application experiencing the fault
- In particular, in mixed criticality systems, failure of a low criticality application must not compromise higher criticality applications
- But the over provision of resources to high criticality tasks could lead to poor schedulability

Constraints of the work

- Uni-processor
- Sporadic task model
- No shared resources/blocking
- No overhead costs
- Fixed Priority Scheduling
- Two Criticality Levels

System Model

- Each task, τ_i, is defined by its period (minimum arrival interval), deadline, computation time and criticality level:
- T_i, D_i, C_i, L_i
- but worst-case computation time is a function of criticality, so:
- T_i, D_i, $\vec{C}_i$, L_i

Criticality Level

- High criticality tasks use WCET estimation techniques that are inheritably more conservative than those for low criticality tasks
- So, $L1 > L2 \Rightarrow C(L1) \geq C(L2)$ for any two criticality levels $L1$ and $L2$

Criticality Level

- Task τ_i with criticality level L_i will have one value from its $\vec{C}_i$ vector that defines its *representative* computation time
- This is the value corresponding to L_i, ie. $C_i(L_i)$
- This will be given the normal symbol C_i

Implementation Schemes

- Partitioned Criticality (PC) – a standard scheme sometimes called *criticality monotonic priority assignment*
- Static Mixed Criticality (SMC)
- Adaptive Mixed Criticality (AMC)

Partitioned

- Priorities are assigned according to criticality, so all jobs of criticality L1 have a higher priority than all jobs of criticality L2 if L1$>$L2
- No run-time monitoring is required
- Poor schedulability

Static - SMC

- All jobs can execute up to their representative execution time C_i (and possible beyond)
- Priorities are assigned (via Audsley's algorithm) to maximise schedulability
- As a result a task with low criticality can have a high priority

Response Time Analysis

For a single criticality system

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

This is solved using standard techniques for recurrence relations

RTA for SMC-NO

If there is no run-time monitoring

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

RTA for SMC

If there **is** run-time monitoring

$$R_i = C_i + \sum_{\tau_j \in \mathbf{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j(min(L_i, L_j))$$

So a low criticality task τ_j only needs its low-crit WCET C_j but is prevented from executing beyond C_j

Adaptive Mixed Criticality (AMC)

- Now if a high-crit task executes for more than its low-crit WCET, all low crit tasks are abandoned
- This significantly increases schedulability
  - By utilising the ‘reserved’ capacity of high-crit tasks

RTA for AMC

Three stages to the analysis:

1. Verifying the schedulability of the LO-criticality mode,
2. Verifying the schedulability of the HI-criticality mode,
3. Verifying the schedulability of the criticality change itself.

Stage 1 - All tasks

$$R_i^{LO} = C_i(LO) + \sum_{j \in \mathbf{hp}(i)} \left\lceil \frac{R_i^{LO}}{T_j} \right\rceil C_j(LO)$$

where $\mathbf{hp}(\mathbf{i})$ is the set of all tasks with priority higher than that of task τ_i.

Stage 2 - HI only

$$R_i^{HI} = C_i + \sum_{j \in \mathbf{hpH}(i)} \left\lceil \frac{R_i^{HI}}{T_j} \right\rceil C_j$$

where $\mathbf{hpH}(i)$ is the set of HI-critical tasks with priority higher than, or equal to, that of task τ_i

Stage 3 - HI only

- Need to analysis the worst-case change from LO to HI behaviour
- Similar problem to mode change analysis
- Worst-case may **not** be when sporadic tasks arrive at their worst-case, hence **tractable analysis** is unlikely to exist

RTS York

Stage 3 - HI only

- Only care about HI-crit tasks
- For HI-crit task τ_i behaviour change must occur between release and the completion time in LO ‘mode’
- So in the interval: $[0, R_i^{LO})$
- We consider two methods for determining sufficient schedulability

RTA for AMC - Method 1

For a HI-crit task, the earlier equation becomes

$$R_i = C_i + \sum_{\tau_j \in \mathbf{hpH}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j + \sum_{\tau_k \in \mathbf{hpL}(i)} \left\lceil \frac{R_i}{T_k} \right\rceil C_k$$

But the final term is bounded by R_i^{LO}, so

RTA for AMC - Method 1

$$R_i^* = C_i + \sum_{\tau_j \in \mathbf{hpH}(i)} \left\lceil \frac{R_i^*}{T_j} \right\rceil C_j + \sum_{\tau_k \in \mathbf{hpL}(i)} \left\lceil \frac{R_i^{LO}}{T_k} \right\rceil C_k$$

We refer to this method as AMC-rtb

RTA for AMC - Method 2

- Here we examine all intermediate points s in $[0, R_i^{LO})$ and take the maximum (interference)
- Only values of s at which a LO-crit task is released need to be considered
- We use analysis that is compatible with Audsley's priority assignment algorithm
- At time s we conservatively assume 'active' HI-crit task consume $C(HI)$ and 'active' LO-crit task complete and use $C(LO)$

Details in the paper

Example

Consider an example task system τ comprised of three tasks, as follows:

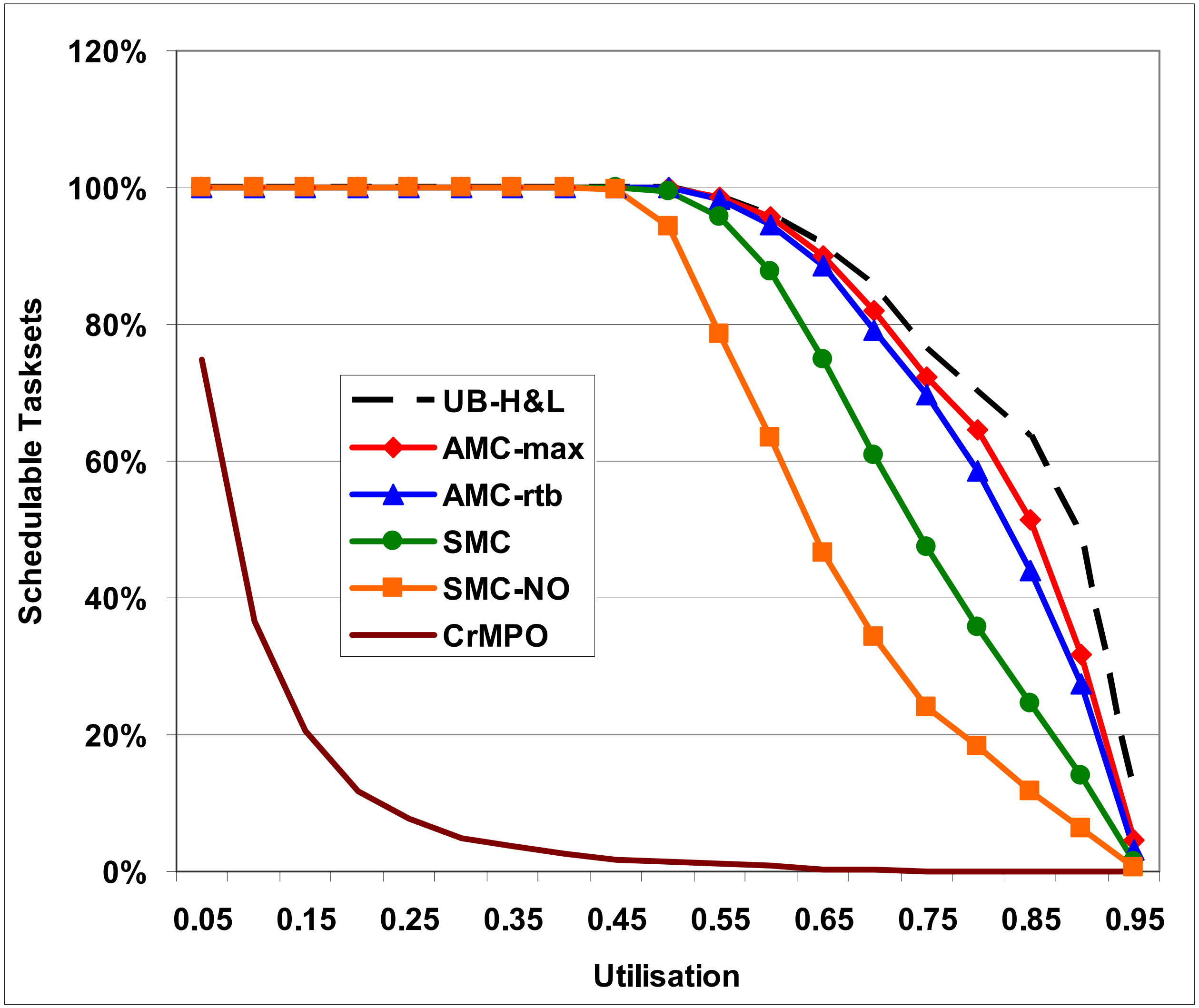
τ_i	L	$C_i(LO)$	$C_i(HI)$	D_i	T_i
τ_1	LO	1	-	2	2
τ_2	HI	1	5	10	10
τ_3	HI	20	20	100	100

Deemed unschedulable by either partitioned priorities or SMC, AMC(1) gives R_3 as 85, AMC(2) gives 59

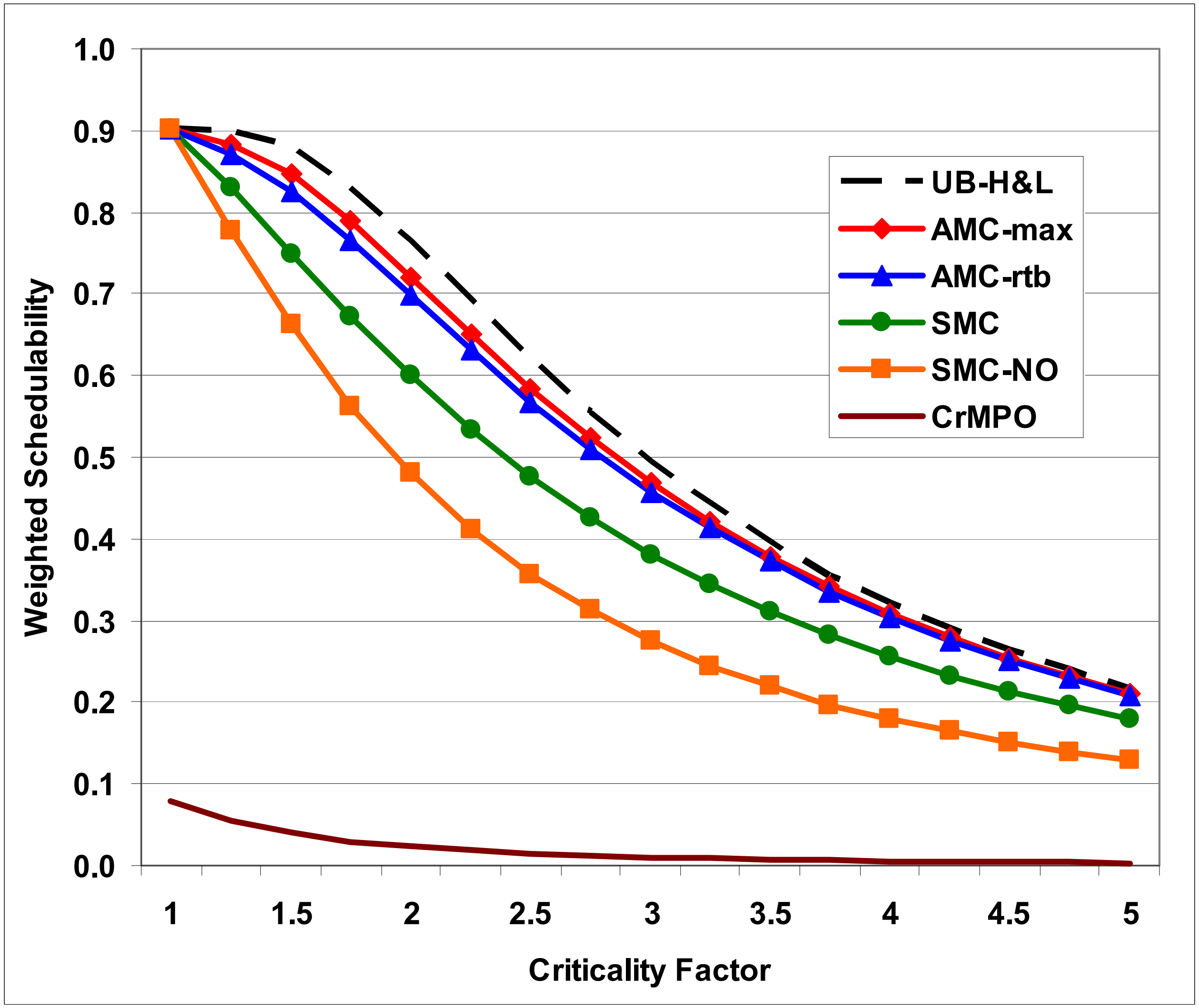
AMC and SMC

- AMC method 2 dominated method 1
- AMC method 1 dominates SMC
- SMC dominates SMC-no
- SMC-no dominated partitioned priorities

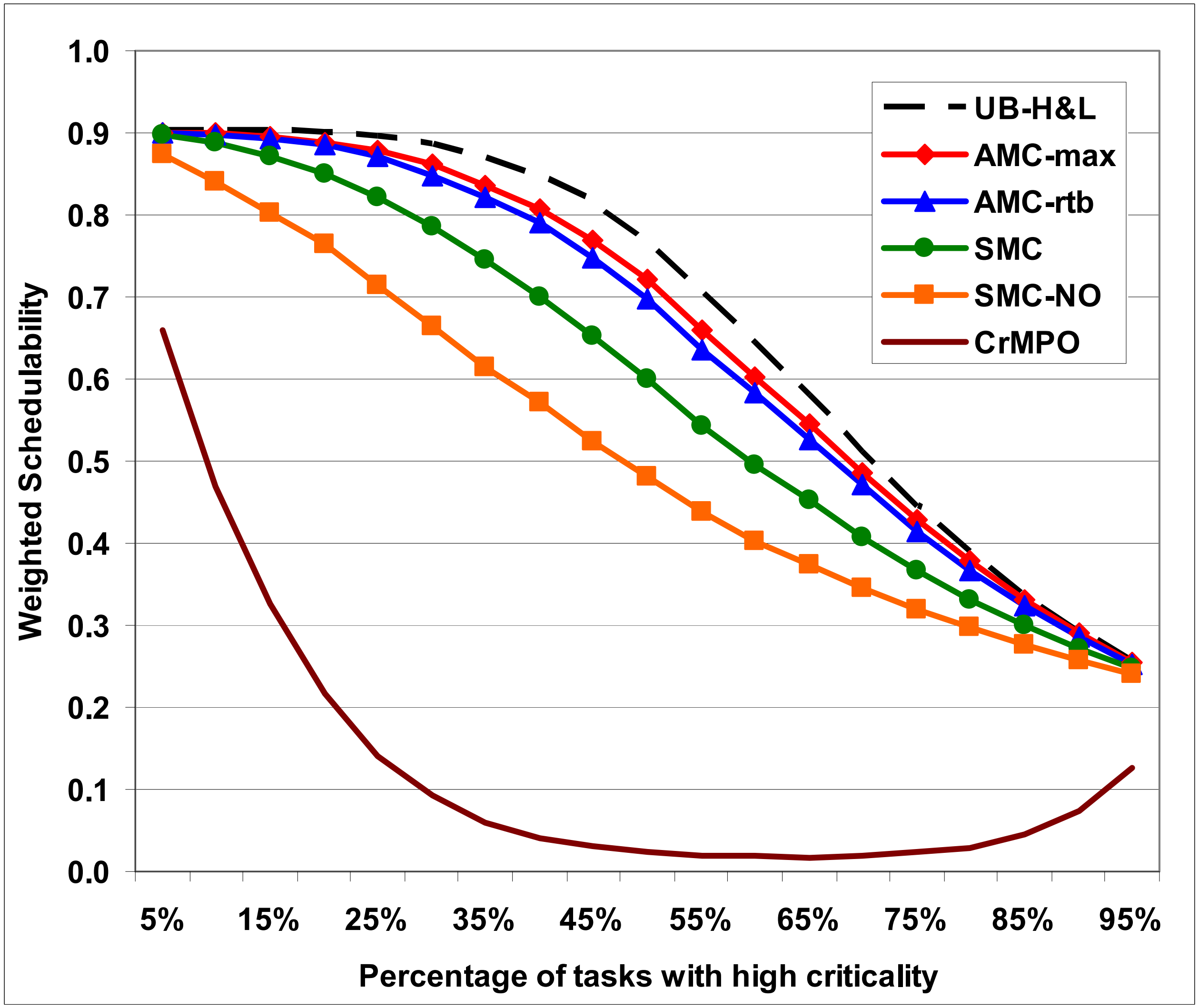
Evaluation: N=20, 50% HI, C*2



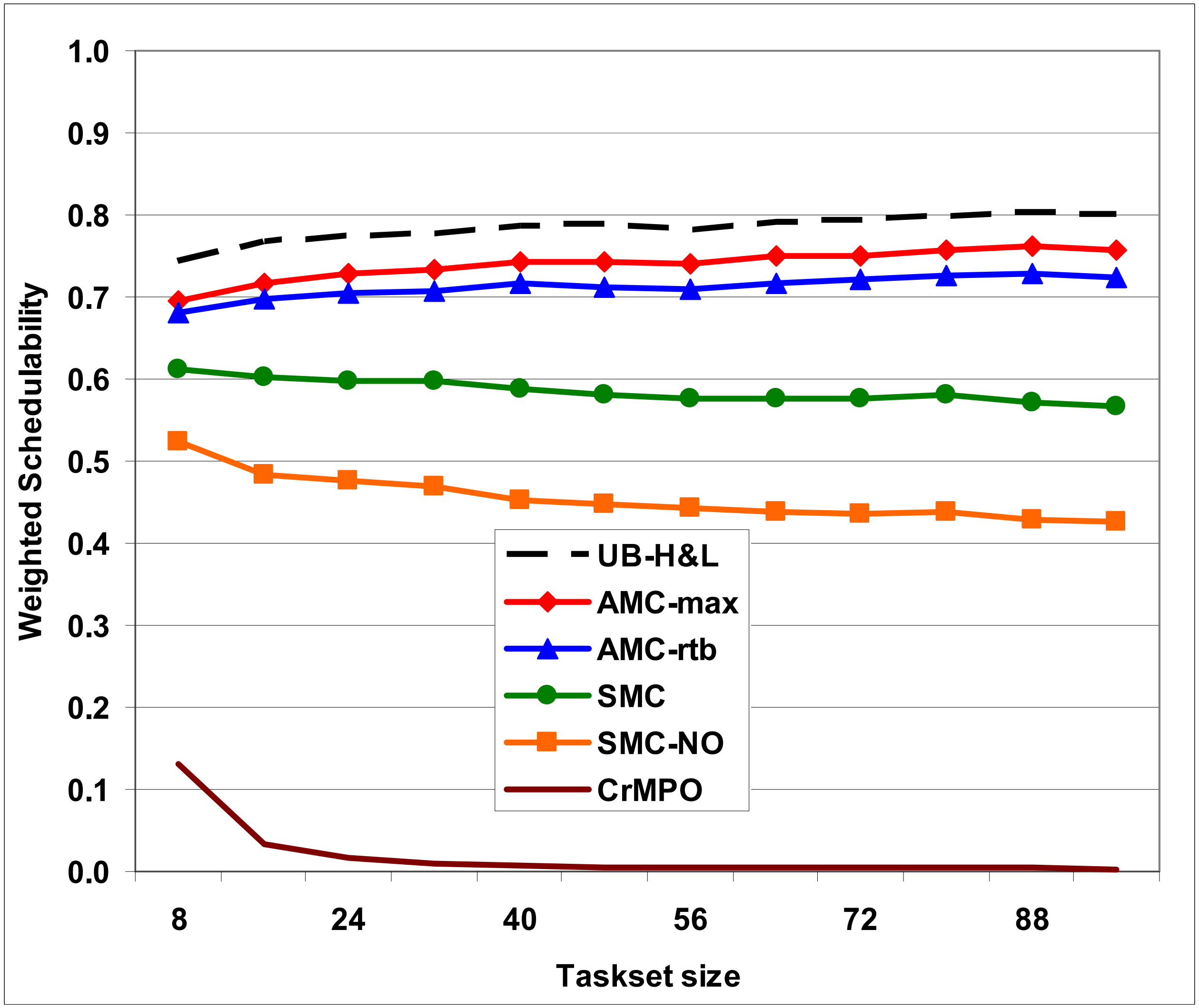
Evaluation: weighted



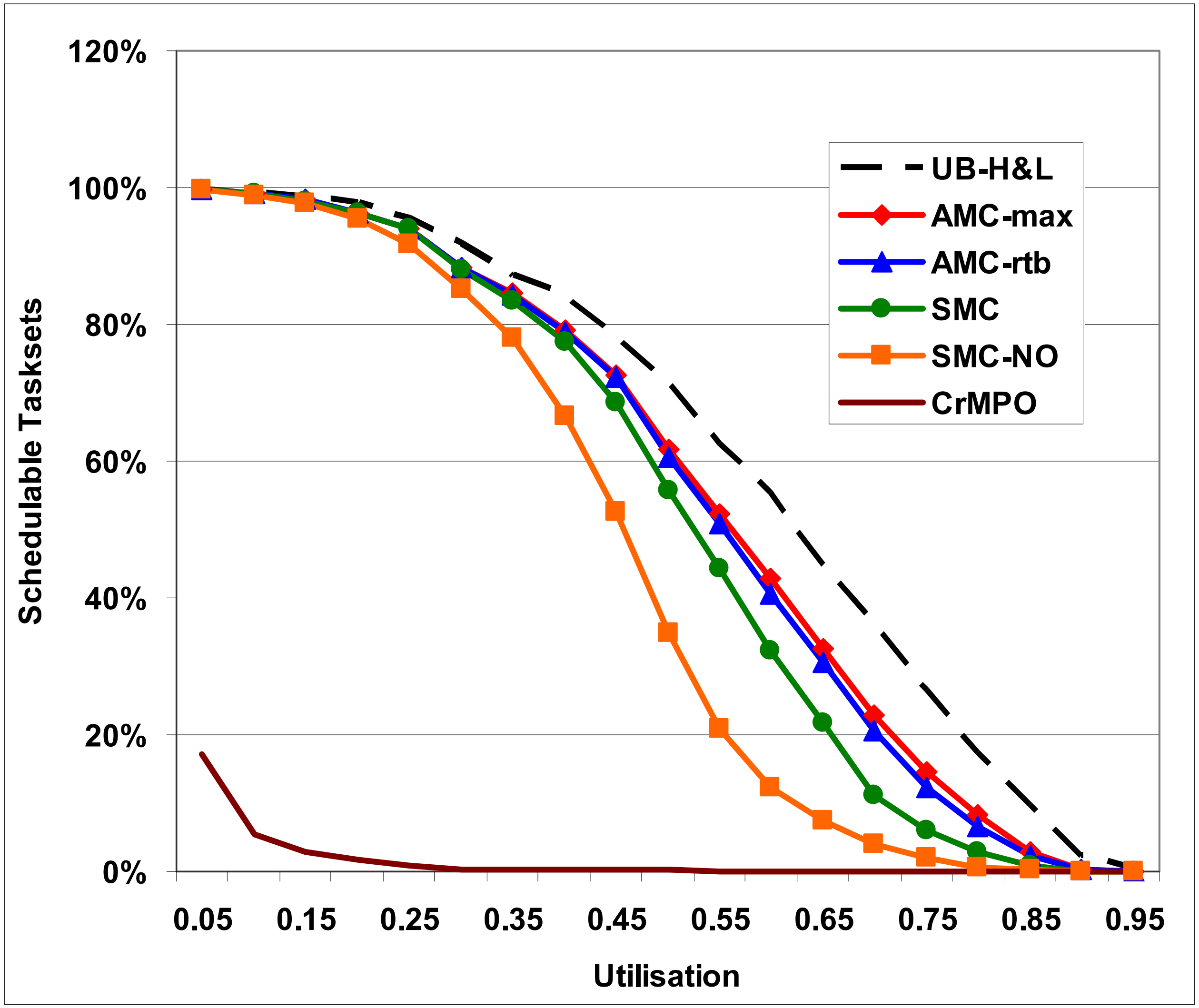
Evaluation: weighted



Evaluation: weighted



Evaluation: D<T



Conclusion

- Mixed Criticality systems are becoming increasingly important
- Smart scheduling can significantly increase resources usage
- The proposed AMC scheme is a significant improvement on SMC
- Simple AMC analysis gets a long way to towards the optimal
- Sensitivity analysis could be used to increase C(LO) values