Dual Priority and EDF: a closer look

Laurent George, Joël Goossens, Damien Masson

Dual Priority and EDF: a closer look

Laurent George, Joël Goossens, Damien Masson

October 19, 2015

1 Introduction

In the context of uniprocessor scheduling, two scheduling algorithms have been very much studied: one in the class of fixed task priority (FTP) where Rate Monotonic (RM) is optimal and one in the class of fixed job priority (FJP), where Earliest Deadline First (EDF) is optimal. RM has the disadvantage of imposing processor utilization less than 100% (i.e., 69% in the worst case) while EDF scheduling can reach 100% of processor utilization.

Some research have been done to overcome this sub-optimality problem. It has been shown that when periods are harmonic, the processor utilization bound of RM is identical to the one of EDF [1]. When no constraint is imposed on the periods, the *dual priority* approach was introduced in 1993 [3]. The scheduler consider two priorities and two phases for each task, each phase has a fixed priority, the transition from a phase to another is made at a fixed time offset from the task release. Dual priority approach is interesting as it is conjectured that a dual priority scheduling can reach the same performances as an EDF scheduler.

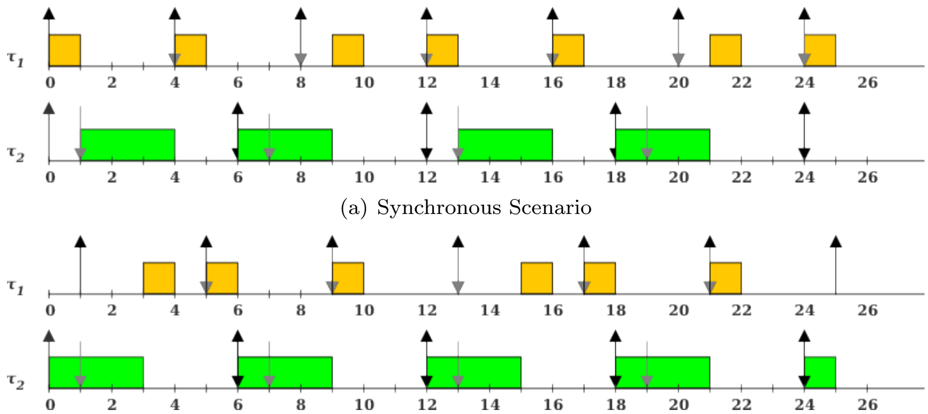
In this paper, we revisit dual priority scheduling for uniprocessor systems with implicit-deadline periodic task set. We recall existing conjectures. Then, we explicit a new class of scheduling FP^k, a fixed priority scheduling that requires at most k promotions at k fixed times. We show that dual priority and EDF scheduling are particular cases of FP^k. Finally, we analyse EDF scheduling trying to study how far it is from a dual priority scheduler in terms of *promotions*.

2 Conjectures and Facts about the Dual Priority Approach

2.1 Conjectures

The following conjecture has been proved only in the case of task sets composed of two-tasks (and remains open in the general case):

Conjecture 1 (Maximal Utilization Bound [2])**.** For any task set with total utilization less than or equal to 100% there exists a dual priority assignment that will meet all deadlines.



(a) Synchronous Scenario

(b) Offset for task τ_1

Figure 1: Counter-intuitive properties illustrations

In addition to Conjecture 1, a conjecture on the priority assignment scheme for dual priority scheduling with RM^2 is as follows:

Definition 1 (RM^2 dual priority scheduling)**.** The phase 1 priorities are RM, the phase 2 priorities are RM with all phase 2 priorities higher than all phase 1 priorities.

Conjecture 2 (Optimality of RM^2)**.** RM^2 priority ordering is optimal for the dual-priority problem.

The following conjecture/open question of Burns will be formalized and closed in Section 3:

Conjecture 3 ([2])**.** At some level (m), 'm-priority' assignment can be made to emulate EDF.

2.2 Facts

We report here (counter-intuitive) properties which illustrate that dual priority scheduling is a scheduling class which differs from FTP and FJP.

Property 1 (Response time of the first job)**.** Consider a *synchronous* implicit-deadline task set, using dual priority the response time of the first job is not necessarily the largest one.

Proof. See [2] Table 1, the response time of second job of τ_2 is larger than the one of the first job. □

Property 2 (The first busy period)**.** Consider a *synchronous* implicit-deadline task set, the first busy period is *not* a feasibility interval using dual priority scheduling.

Proof. Consider the following dual-priority task set (including the promotion time and RM^2 for the priorities): $\tau_1 = (C_1 = 1, T_1 = 4, S_1 = 4, P_1 = 3, P'_1 = 1), \tau_2 = (3, 6, 1, 4, 2)$ where S_i is the promotion deadline, P_i is the initial priority and is P'_i the promoted priority of τ_i. Figure 1(a) shows that the response time of the third job of τ_1 is larger than the previous ones after an idle processor period in the interval $[5, 6]$. □

Property 3 (No critical instant)**.** Considering dual priority scheduling, the synchronous case is not the worst case.

Proof. Consider the same dual priority task set. If we add an offset of 1 for the first release to τ_1, its worst case response time is 3 (see Figure 1(b)), but it was 2 in the synchronous case (See Figure 1(a)). Note that we know these are the worst case response times because a cyclic pattern appears at time 12 in both cases. □

3 The FP^k Algorithm Class

Definition 2 (FP^k Algorithm Class)**.** The FP^k scheduling is a generalization of the dual-priority scheme, the task characteristics $\vec{s_i}$ and $\vec{p_i}$ are arrays in this generalization. More formally, each task $\tau_i = (O_i, C_i, T_i = D_i, \vec{s_i}, \vec{p_i})$ where O_i, C_i, T_i, D_i are popular Liu and Layland task parameters and $\vec{s}$, $\vec{p}$ are two vectors of k integers. A FP^k algorithm assigns a priority to each job of task τ_i exactly k times, at relative (to the jobs release) time instants in $\vec{s_i}$ with the corresponding priority levels in $\vec{p_i}$.

3.1 RM, EDF and Dual-Priority are FP^k Algorithms

FTP and consequently RM are obviously FP^1 schedulers, dual priority is obviously an FP^2 scheduler. The next result answers to Conjecture 3.

Property 4. EDF is an $\mathsf{FP}^{\max_{i=1,\ldots,n} D_i}$ scheduling.

Proof. For task τ_i, we have $\vec{s_i} = \{0, 1, 2, 3, \ldots, \max_{i=1,\ldots,n} D_i\}$ and $\vec{p_i} = \{D_i, D_i - 1, D_i - 2, \ldots, D_i - k + 1\}$. Note that theoretically the priority can be negative if the task-set is not schedulable. □

4 Promotion Point Study with EDF

4.1 Definitions

Definition 3 ($\mathsf{HP}(\tau_i, t)$)**.** For any priority-driven scheduler we denote by $\mathsf{HP}(\tau_i, t)$ the set of task indexes (among all the n tasks active or not) which have a higher priority than the current job of τ_i at time instant t.

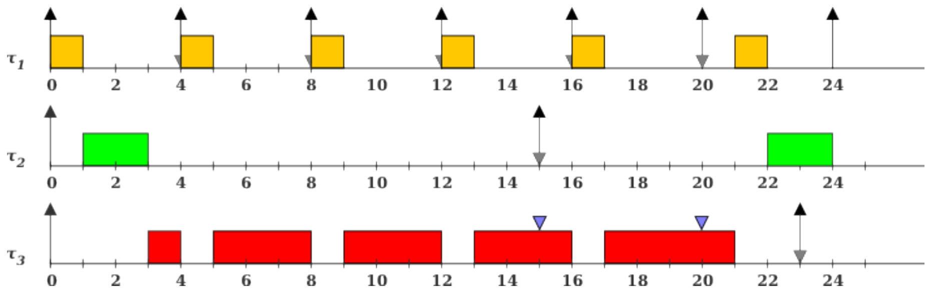


Figure 2: EDF Schedule of Synchronous Periodic Task Set $\tau_1 = (C_1 = 1, T_1 = 4), \tau_2 = (2, 15), \tau_3 = (14, 23)$

Definition 4 (Job Promotion at time t)**.** Let us consider a job J of task τ_i *running* at time t, we say that job J is promoted at time t if $\exists$ instant $\ell < t$ such that (i) the very same job J is running at time ℓ, (ii) $\mathsf{HP}(\tau_i, t) \subset \mathsf{HP}(\tau_i, \ell)$ and (iii) $\nexists s$ s.t. $\ell \leq s < t$ with $\mathsf{HP}(\tau_i, t) = \mathsf{HP}(\tau_i, s)$.

Lemma 1. In an EDF schedule, the promotions point of a given job can only occur at other jobs release times.

Proof. Less formally, Definition 4 states that a promotion occurs when the relative priority of two tasks τ_a and τ_b are permuted during the execution of one job of τ_b. Since EDF is a job-level fixed priority algorithm, this can only happen at the release time of a job. □

Lemma 2. In an EDF schedule, when a job J_a is promoted by the activation of a job J_b, J_b is the last job of τ_b before the next release of τ_a.

Proof. If it exists an other job of τ_b activated before the next release of τ_a, this implies that the deadline of J_b, which coincides with this activation ($\forall i, D_i = T_i$), is lesser than the deadline of J_a, and there is no promotion. □

4.2 EDF may needs more than one promotion per Job

By definition we know that FTP schedulers do not promote any job. Similarly by definition we know that dual priority schedulers do promote each job *at most* once. The next example shows that, unfortunately, EDF may require to promote a job strictly more than one time.

Example 1. Consider the following synchronous periodic task set composed of 3 tasks: $\tau_1 = (C_1 = 1, T_1 = 4), \tau_2 = (2, 15), \tau_3 = (14, 23)$. The EDF schedule (see Figure 2) shows that task τ_3 is promoted twice: initially, at time $t = 0$, $\mathsf{HP}(\tau_3, 0) = \{2, 1\}$ while $\mathsf{HP}(\tau_3, 15) = \{1\}$ and $\mathsf{HP}(\tau_3 = 20) = \emptyset$. Consequently we have *two* promotions at time $t = 15$ and $t = 20$.

4.3 Statistical Results

We perform simulations other 40000 randomly generated task sets. Each system is composed by 10 tasks with processor utilization varying from 0 to 1. The system repartition regarding the maximum number of promotions EDF uses per jobs is as follow: 16329 systems with 0 promotion, 19739 with 1, 3499 with 2, 393 with 3, 38 with 4, 2 with 5. Note that this values may be reduced with different tie-deadline break rules. Note also that the more the processor utilization is, the most likely it is to have promotions. We do not detail results here due to space limitation. Moreover, it seems that cases where EDF needs more than one promotion are quite rare and a deap study of these cases may be an interesting way to try to prove Conjecture 1.

5 Conclusions and Future Work

In this paper, we revisit the dual priority problem that conjecture that RM^2 is optimal for the dual priority problem . We explicit the FP^k class of scheduling requiring k promotions at fixed times. We show that RM^2 and EDF are particular cases of FP^k and provide counter-intuitive results concerning dual-priority scheduling. As a future work, we would like to propose an optimal promotion time algorithm for FP^k scheduling.

References

[1] Bonifaci, V., Marchetti-Spaccamela, A., Megow, N., and Wiese, A. Polynomial-time exact schedulability tests for harmonic real-time tasks. In *RTSS 13*, pp. 236–245.

[2] Burns, A. Dual priority scheduling: Is the processor utilisation bound 100%. In *RTSOPS 10* (2010).

[3] Burns, A., and Wellings, A. Dual priority assignment: A practical method for increasing processor utilisation. In *ECRTS 93*, pp. 48–53.