# Utilization Bounds on Allocating Rate-Monotonic Scheduled Multi-Mode Tasks on Multiprocessor Systems

Wen-Hung Huang
Department of Computer Science
TU Dortmund University, Germany
wen-hung.huang@tu-dortmund.de

Jian-Jia Chen
Department of Computer Science
TU Dortmund University, Germany
jia.chen@tu-dortmund.de

## ABSTRACT

Formal models used for representing recurrent real-time processes have traditionally been characterized by a collection of jobs that are released periodically. However, such a modeling may result in resource under-utilization in systems whose behaviors are not entirely periodic. For instance, tasks in cyber-physical system (CPS) may change their service levels, e.g., periods and/or execution times, to adapt to the changes of environments. In this work, we study a model that is a generalization of the periodic task model, called multi-mode task model: a task has several modes specified with different execution times and periods to switch during runtime, independent of other tasks. Moreover, we study the problem of allocating a set of multi-mode tasks on a homogeneous multiprocessor system. We present a scheduling algorithm using any reasonable allocation decreasing (RAD) algorithm for task allocations for scheduling multi-mode tasks on multiprocessor systems. We prove that this algorithm achieves 38% utilization for implicit-deadline rate-monotonic (RM) scheduled multi-mode tasks on multiprocessor systems.

## 1. INTRODUCTION

Many real-time systems are modeled as a finite collection of independent recurrent tasks, each of which generates infinite jobs repetitively. Traditionally, each task is characterized by its worst-case execution time (WCET), its period/minimum inter-arrival time, and its relative deadline, as known as periodic/sporadic tasks.

**The mode change model** Such a sporadic task model has been thoroughly studied in real-time community since 1970's. Traditionally, the emphasis of designing embedded systems is on providing high computational components but obliviously on the link between the computational and physical components. The recent trend towards models adaptable to the changes of environments has given rise to a new aspect of implementing real-time systems. For example, tasks in automotive systems are linked to rotation (e.g., of the crankshaft, gears, or wheels). Their activation rate is proportional to the angular velocity of a specific device, which in turn determines the execution mode to be invoked. Therefore, it is encouraged to design a real-time task having several modes to switch during runtime, called *multi-mode* task.

This paper is supported by DFG, as part of the Collaborative Research Center SFB876 (http://sfb876.tu-dortmund.de/).



*DAC '16, June 05 - 09, 2016, Austin, TX, USA*



ISBN 978-1-4503-4236-0/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2897937.2898108

The mode change model is a generalization of the sporadic task model. Each multi-mode task $\tau_i$ with $H_i$ modes is denoted by a set of triplets: $\tau_i = \{\tau_i^1 = (C_i^1, T_i^1, D_i^1), \tau_i^2 = (C_i^2, T_i^2, D_i^2), \ldots, \tau_i^{H_i} = (C_i^{H_i}, T_i^{H_i}, D_i^{H_i})\}$, where $C_i^h$ denotes the *worst-case execution time* (WCET) of task $\tau_i$ under mode $h$; $T_i^h$ denotes the *minimum inter-arrival time* of task $\tau_i$ under mode $h$; and $D_i^h$ denotes the *relative deadline.* The importance of mode changes for real-time systems has been pointed out in many perspectives, for instance, automotive Electronic Control Units (ECU) [3,6,9] and server-based systems [10] in uniprocessor systems. Some of these researches explicitly study the potential mode changes imposed by the physical environment under the names of adaptive variable-rate (AVR) model [3] or variable-rate dependent behaviour (VRB) model [6], etc. Some model the mode changes with constrained or unconstrained invoked sequences like the generalized multiframe (GMF) model [2], acyclie task model [1], digraph real-time (DRT) model [12], and multi-mode task model [9].

**Multiprocessor systems** Due to the issue of power consumption and excessive heat dissipation in increasing high processor clock speeds, there has been a move towards using multiprocessor platforms. To schedule real-time tasks on multiprocessor platforms, there are two widely adopted approaches: *partitioned* and *global* scheduling. In *global* scheduling, a global queue is implemented for all instances of tasks, namely jobs, and the jobs can be arbitrarily assigned to any processor, depending upon a global scheduling scheme. Hence, different jobs of the same task may execute upon different processors. In *partitioned* scheduling, contrary to global scheduling, each task is assigned statically to one processor and cannot migrate to the others. As a result, all jobs generated by a task only execute on the processor where the task is assigned. In this paper, we consider the partitioned strategy, and we assume that all the tasks allocated to a processor are preemptively scheduled using fixed priority scheduling.

In the sporadic task model, when the relative deadline is equal to the period, the task allocation for partitioned scheduling on multiprocessor systems is analogous to the well-known *bin packing problem* [8]. In the bin packing problem, the objective is to pack a set of items into a number of bins such that the volume of the packed items in every bin does not exceed that of the bin and the number of used bins is minimized.

**The siginificance of this work.** In this work, we study how to allocate implicit-deadline multi-mode tasks (i.e., $D_i^h = T_i^h$) onto multiprocessor systems. The objective of this paper is to obtain the utilization bound for multi-mode tasks on $M$ identical multiprocessor systems by using the simple allocation algorithms combined with the schedulability condition in uniprocessor, where the utilization of a multi-mode task $\tau_i$ is $U_i = \max_{h=1,2,\ldots,H_i} C_i^h/T_i^h$. The contributions are summarized as follows:

- Following the result in [9], we can trivially provide a utilization bound $\sum_i U_i \leq \frac{2-\sqrt{2}}{2}M \approx 0.293 \cdot M$ if $U_i \leq 2-\sqrt{2}$ for every task $\tau_i$. More favorably, a utilization bound $\sum_i U_i \leq \frac{3-\sqrt{5}}{2}M \approx 0.381 \cdot M$ on multiprocessor systems under RM scheduling is derived, in Section 4.
- In Section 5, we then further improve this bound by considering the upper bound on the utilization for every task.
- We explicitly also conclude that the mode change model studied in this paper is a relaxation[1] of the generalized multiframe (GMF) model [2], the variable-rate dependent behavior (VRB) model [6] (a.k.a the adaptive variable-rate (AVR) model [3,4]), and the digraph real-time (DRT) model [12]. As a consequence, we also derived utilization bounds for these models on multiprocessor systems, in Section 6.

To the best of our knowledge, this is the first work to assign multi-mode real-time tasks on multiprocessor systems.

## 2. RELATED WORK

The problem of scheduling sporadic real-time tasks on multiprocessors regarding partitioned scheduling has been addressed in a number of studies. Please refer to [5] for an extensive survey of the research that has been conducted within the real-time scheduling community on multiprocessors scheduling problems. Unfortunately, because the studied mode change model is a generalized model of the sporadic task model, these results above are by no means applicable.

Regarding the studied mode change model, it has been recently shown in [9] that the utilization bound $2-\sqrt{2}$ can be derived for a system with implicit-deadline multi-mode tasks if each mode is prioritized according to rate-monotonic (RM) scheduling. In other words, this bound represents the maximum utilization of a system for multi-mode tasks before any task can miss its deadline. There are several models related to the studied mode change model, for instance, the generalized multiframe (GMF) model [2], the acyclie task model [1], the digraph real-time model (DRT) [12], the adaptive variable-rate (AVR) model [3,4], and the variable rate-dependent behavior (VRB) task model [6]. The generalized multiframe (GMF) model [2] allows a task to cycle through a static list of job types, each with potentially different WCET bounds and relative deadlines. Stigge et al. [12] propose a more expressive model, called digraph real-time model (DRT), in which the release structures of different types of jobs are represented by a directed graph. The acyclic task model [1] allows a job to have any arbitrary execution time under the assumption that the absolute deadline and the arrival time of the next job of this task are both after the arrival time of the job plus the utilization times the execution time of the job. In the variable rate-dependent behavior (VRB) model (a.k.a. adaptive varying-rate (AVR) model) [3,4,6], tasks are linked to rotation (e.g., of the crankshaft, gears, or wheels). Their activation rate is proportional to the angular velocity of a specific device, which in turn determines the execution mode to be invoked.

To the best of our knowledge, there is no result regarding the above mode change models on multiprocessor systems.

## 3. SYSTEM AND TASK MODEL

We assume in this paper that we have a multiprocessor platform comprised of $M$ identical processors, on which $N$ multi-mode tasks are scheduled. We further assume that $N > M$, since the case with $N \leq M$ is trivial. We restrict our attention on *partitioned scheduling*: each task is statically assigned onto one processor. We consider the multiprocessor system to execute a set of $N$ independent, preemptive, real-time, multi-mode tasks $\tau = \{\tau_1, \tau_2, ..., \tau_N\}$. A multi-mode task $\tau_i$ with $H_i$ modes is denoted by a set of triplets:

$$\tau_i = \{\tau_i^1 = (C_i^1, T_i^1, D_i^1), \tau_i^2 = (C_i^2, T_i^2, D_i^2), ..., \tau_i^{H_i} = (C_i^{H_i}, T_i^{H_i}, D_i^{H_i})\}$$

For the task mode $\tau_i^h$, $C_i^h$ denotes the worst-case execution time, $T_i^h$ denotes the minimum inter-arrival time, and $D_i^h$ denotes the relative deadline of task mode $\tau_i^h$. When a job of mode $\tau_i^h$ is released at time $t$, the next release time of jobs of task $\tau_i$ is no earlier than $t + T_i^h$, and when a job of mode $\tau_i^h$ is released at time $t$, this job has to be finished no later than its *absolute deadline* at time $t + D_i^h$. A multi-mode task can switch the mode being executing from one to another only if the temporal constraint is met. That is, assuming that a job of mode $\tau_i^h$ is released at time $t$, the earliest time to switch to other modes is $t + T_i^h$. Note that the concept of the mode change model studied in this paper is distinct from that of system-wide operating modes [7,11,13]. Our studied model characterizes the system where different *tasks* may progress through their own execution modes independent of each other.

Throughout this paper, we restrict ourselves on *implicit-deadline* ($D_i^h = T_i^h$) multi-mode task systems. The ratio $C_i^h/T_i^h$ (worst-case execution time to the minimum inter-arrival time) denotes the utilization factor $U_i^h$ of mode $\tau_i^h$. We denote the (maximum) utilization of task $\tau_i$ as $U_i = max_{h=1,...,H_i}\{U_i^h\}$. We assume $0 < U_i \leq \alpha \leq 1$, where $\alpha$ is the upper bound on utilizations for every task. We further assume $U^{sum} = \sum_{i=1}^{N} U_i \leq M$.

We consider in this paper that the tasks are scheduled under fixed-priority scheduling in *mode-level*: all the jobs generated by the same task mode have the same priority; however, jobs from different task modes may have different priority levels. We specifically assume the priority for each mode is assigned according to *rate-monotonic* (RM) scheduling: the smaller the minimum inter-arrival time (also referred to as the period historically), the higher the priority level. We assume that the system is fully *preemptive* and that the cost of preemption has been subsumed into the worst-case execution time of each mode.

## 4. MULTIPROCESSOR UTILIZATION

### 4.1 Preliminaries

To the best of our knowledge, the only result related to the utilization bound on uniprocessor systems for multi-mode rate-monotonic scheduled tasks is that recently given by Huang and Chen [9]. A uniprocessor system comprised of a set of multi-mode rate-monotonic scheduled tasks is schedulable if

$$U^{sum} \leq 2 - \sqrt{2} \tag{1}$$

This also aligns with the utilization bound of the acyclic task model by Abdelzaher et al. [1]. In addition to the above total utilization bound, a more precise utilization bound based a quadratic bound (denoted as QB for the rest of this paper) is also presented in [9]. For completeness, we state this bound in the following lemma:

Lemma 1 (Huang and Chen [9]). *A multi-mode task set $\Gamma$ with implicit deadline is schedulable on a uniprocessor under RM scheduling if $\sum_{\tau_i \in \Gamma} U_i \leq 1$ and*

$$U_a \leq 1 - 2\sum_{\tau_i \in \Gamma \setminus \{\tau_a\}} U_i + \frac{1}{2}\left(\sum_{\tau_i \in \Gamma \setminus \{\tau_a\}} U_i\right)^2 + \frac{1}{2}\sum_{\tau_i \in \Gamma \setminus \{\tau_a\}} (U_i)^2 \tag{2}$$

*where $U_a = min_{\tau_i \in \Gamma} U_i$.*

[1]This is under certain assumptions to be explained in Section 6.

PROOF. This comes from Theorem 5 in [9]. □

Note that the above lemma requires to test both $\sum_{\tau_i \in \Gamma} U_i \leq 1$ and Eq. (2). Testing only Eq. (2) is unsafe since the quadratic form in the right-hand side of Eq. (2) may become larger than 1 when $\sum_{\tau_i \in \Gamma \setminus \{\tau_a\}} U_i$ is sufficiently large. However, it is still possible to only test a variance of Eq. (2) if we test the tasks in $\Gamma$ sequentially, which will be used in our algorithm and analysis.

THEOREM 1. *Suppose that the tasks in $\Gamma$ are sorted and indexed such that $U_1 \geq U_2 \geq \cdots \geq U_{|\Gamma|}$. A multi-mode task set $\Gamma$ with implicit deadline is schedulable on a uniprocessor under RM scheduling if $\forall k = 1, ..., \Gamma$,*

$$U_k \leq 1 - 2\sum_{i=1}^{k-1} U_i + \frac{1}{2}\left(\sum_{i=1}^{k-1} U_i\right)^2 + \frac{1}{2}\sum_{i=1}^{k-1} (U_i)^2 \tag{3}$$

PROOF. When $k$ is 1, this holds naturally. In this proof, we show that any feasible allocation by using Eq. (3) implies that $\sum_{i=1}^{k} U_i \leq 1$ under the assumption that $\sum_{i=1}^{k-1} U_i \leq 1$ for $k = 2, 3, \ldots, |\Gamma|$.

Due to the evidence that $\left(\sum_{i=1}^{k-1} U_i\right)^2 \geq \sum_{i=1}^{k-1} (U_i)^2$, the satisfaction of Eq. (3) implies that

$$U_k \leq 1 - 2\sum_{i=1}^{k-1} U_i + \frac{1}{2}\left(\sum_{i=1}^{k-1} U_i\right)^2 + \frac{1}{2}\left(\sum_{i=1}^{k-1} U_i\right)^2 \tag{4}$$

Let $z$ denote $\sum_{j=1}^{k-1} U_i$. Replacing $\sum_{i=1}^{k-1} U_i$ by $z$, we have $U_k \leq 1 - 2z + z^2$. It follows that

$$U_k + z \leq 1 - z + z^2 \tag{5}$$

Hence, the total utilization $U_k + z$ of the $k$ tasks to pass Eq. (3) is upper bounded by the right-hand side of Eq. (5), which is upper bounded by 1 if $0 \leq z \leq 1$. By the assumption that $U_i \leq 1$ for every task $\tau_i \in \Gamma$ and $0 \leq z \leq 1$, if Eq. (3) holds for every $k = 1, 2, \ldots, |\Gamma|$, the task set $\Gamma$ can also pass the test in Lemma 1. □

## 4.2 Reasonable Allocation Decreasing Algorithm

In this section, we first introduce some simple heuristic allocation algorithms that have been developed for the bin-packing problem [8]. An allocation algorithm is said to be *reasonable* if the tasks are allocated sequentially and the allocation algorithm fails to allocate a task to the previously allocated processors only if there is no processor having sufficient *capacity* to hold the pending task. There are three simple heuristic reasonable allocation (RA) algorithms, known as First-Fit (FF), Best-Fit (BF), and Worst-Fit (WF). The First-Fit algorithm places the item in the first bin that can accommodate the item. If no bin is found, it opens a new bin and puts the item to the new bin. The Best-Fit (Worst-Fit, respectively) algorithm places each item to the bin with the *lowest* (*largest*, respectively) remaining capacity among all the bins with sufficient capacity to accommodate the item.

In addition, if the items are ordered by a decreasing *ratio* of the value to the weight, of an item, before allocation, it is known as a *reasonable allocation decreasing* (RAD) algorithm, described in [8]. A RAD algorithm is characterized as follows:

- Items are ordered by a decreasing *ratio* of the value to the weight before allocation.
- Items are allocated sequentially by following the order defined above and the allocation algorithm fails only if there is no bin having sufficient capacity to hold the pending item.

It has been shown that the approximation factor of an RAD algorithm is 2 for the bin packing problem [14, Chapter 9]. More precisely, if an RAD algorithm allocates $M$ bins with $M \geq 2$, the overall weight is strictly larger than $\frac{M}{2}$ times of the bin size.[2] Therefore, we can directly use the utilization bound $2 - \sqrt{2}$ as the bin size and reach the following theorem.

**Algorithm 1:** First-Fit Decreasing (FFD)

**input** : A set $\tau$ of multi-mode tasks and $M$ identical processors
**output**: Task allocations $\Gamma_j$ and the feasibility of system $\tau$
sort the given $N$ tasks in $\tau$ s.t. $U_1 \geq U_2 \geq \cdots \geq U_N$;
$\Gamma_j \leftarrow \emptyset, \forall j = 1, 2, \ldots, M$;
**for** $k = 1, 2, \ldots, N$ **do**
  **for** $j = 1, 2, \ldots, M$ **do**
    **if**
    $U_k \leq 1 - 2\sum_{\tau_i \in \Gamma_j} U_i + \frac{1}{2}(\sum_{\tau_i \in \Gamma_j} U_i)^2 + \frac{1}{2}\sum_{\tau_i \in \Gamma_j} (U_i)^2$
    **then**
      $\Gamma_j \leftarrow \Gamma_j \cup \{\tau_k\}$; `// assign` $\tau_k$ `to proc.` $j$
      break (continue the outer loop) ;
  return "infeasible allocation";
return "feasible allocation";

THEOREM 2 (RAD-TUB). *An implicit-deadline multi-mode system $\tau$ is feasible under RM scheduling and a reasonable allocation decreasing (RAD) algorithm using the total utilization bound $2 - \sqrt{2}$ as the bin size if*

$$U_i \leq 2 - \sqrt{2} \quad \forall \tau_i \in \tau \text{ and } U^{sum} \leq (\frac{2 - \sqrt{2}}{2})M \cong 0.293 \cdot M \tag{6}$$

PROOF. This comes from the above discussions and the link to the bin packing problem. The condition $U_i \leq 2 - \sqrt{2}$ for every task $\tau_i$ is needed since RAD algorithms assume that the size of any item is smaller than the bin size. □

Using the total utilization bound is a simple solution, but we will show in the next subsection that using the quadratic bound (QB) in Theorem 1 can reach much better results.

## 4.3 Utilization Bound by Using QB

We consider any reasonable allocation decreasing algorithm using the quadratic bound (QB) in Theorem 1, for example, First-Fit Decreasing (FFD) algorithm using QB (also see Algorithm 1). Note that by using the RAD algorithm, when we consider to assign task $\tau_k$, we know that the utilization of $\tau_k$ is no more than the utilization of the tasks that have been assigned onto the processors.

For the rest of the analysis, we assume that the $N$ given tasks are sorted such that $U_1 \geq U_2 \geq \cdots \geq U_N$. Let $\tau_{Y+1}$ be the task that fails to be assigned to any of the processors, as $Y$ tasks have been assigned onto the processors successfully. Let $\Gamma_j$ be the set of the tasks that have been assigned on processor $j$ before task $\tau_{Y+1}$ is considered. Thus, by Theorem 1, the schedulability condition for each processor that cannot accommodate task $\tau_{Y+1}$ can be concluded as follows:

$$\forall j \in [1, M], \;\; U_{Y+1} > 1 - 2\sum_{\tau_i \in \Gamma_j} U_i + \frac{1}{2}(\sum_{\tau_i \in \Gamma_j} U_i)^2 + \frac{1}{2}\sum_{\tau_i \in \Gamma_j} (U_i)^2$$

Notice that $\sum_{j=1}^{M} |\Gamma_j| = Y$. By adding these $M$ inequalities together, we therefore have that

$$MU_{Y+1} > M - 2\sum_{i=1}^{Y} U_i + \frac{1}{2}\sum_{i=1}^{Y} (U_i)^2 + \frac{1}{2}\left((\sum_{\tau_i \in \Gamma_1} U_i)^2 + (\sum_{\tau_i \in \Gamma_2} U_i)^2 + \ldots + (\sum_{\tau_i \in \Gamma_M} U_i)^2\right) \tag{7}$$

[2]Although the description in the book [14] is for the first-fit algorithm, the factor 2 holds also for any RAD algorithm. The description in [14] was for a trivial analysis with overall weight strictly larger than $\frac{M-1}{2}$. The overall weight $\frac{M}{2}$ can be easily achieved as well.

Here, we can use Cauchy's Inequality $\sum_i r_i^2 \sum_i s_i^2 \geq (\sum_i r_i s_i)^2$ where all of $r_i, s_i \in \mathbb{R}$. When $s_i$ is replaced by 1, we have $x \sum_i r_i^2 \geq (\sum_i r_i)^2$, where $x$ is the number of terms of $i$ in the summation. By this inequality, we then have that

$$\left( (\sum_{\tau_i \in \Gamma_1} U_i)^2 + (\sum_{\tau_i \in \Gamma_2} U_i)^2 + ... + (\sum_{\tau_i \in \Gamma_M} U_i)^2 \right) \geq \frac{1}{M} (\sum_{i=1}^{Y} U_i)^2. \tag{8}$$

Consequently, it follows that by Eq. (7) and (8)

$$MU_{Y+1} > M - 2\sum_{i=1}^{Y} U_i + \frac{1}{2}\sum_{i=1}^{Y} (U_i)^2 + \frac{1}{2M}(\sum_{i=1}^{Y} U_i)^2 \tag{9}$$

In the following lemma, we first provide a necessary condition for an RAD algorithm to fail to allocate a task onto the $M$ processors:

LEMMA 2. *Let $Y$ be the number of multi-mode tasks that have been assigned feasibly on $M$ processors. If a reasonable allocation decreasing (RAD) algorithm using quadratic bound (QB) fails to allocate task $\tau_{Y+1}$, then the following condition must hold:*

$$U^{sum} > \sum_{i=1}^{Y} U_i > \frac{1 + 2\gamma - \sqrt{1 + 2\gamma + 2\gamma^2}}{(1+\gamma)} \cdot M \tag{10}$$

*where $\gamma = \frac{Y}{M}$.*

PROOF. We have shown that the condition in Eq. (9) is necessary for an RAD algorithm using QB to fail to allocate a task onto the $M$ processors. Due to the non-increasing utilization ordering of the tasks, we have

$$\sum_{i=1}^{Y} U_i \geq YU_{Y+1} \tag{11}$$

Our objective in this proof is to find the infimum $\sum_{i=1}^{Y} U_i$ such that Eq. (9) and (11) always hold. This is equivalent to the following quadratic programming (QP), where $U_i$s are variables:

$$\text{min.} \quad \sum_{i=1}^{Y} U_i \tag{12a}$$

$$\text{s.t.} \quad MU_{Y+1} \geq M - 2\sum_{i=1}^{Y} U_i + \frac{1}{2M}(\sum_{i=1}^{Y} U_i)^2 + \frac{1}{2}\sum_{i=1}^{Y}(U_i)^2 \tag{12b}$$

$$\sum_{i=1}^{Y} U_i \geq YU_{Y+1} \tag{12c}$$

Let $\lambda$ be the multiplier of the schedulability constraint by Eq. (12b) and $\mu$ be the multiplier of the constraint by Eq. (12c). The Lagrange function is

$$L(U_1, U_2, ..., U_Y) = \sum_{i=1}^{Y} U_i - \mu(\sum_{i=1}^{Y} U_i - YU_{Y+1}) -$$

$$\lambda \left( MU_{Y+1} - M + 2\sum_{i=1}^{Y} U_i - \frac{1}{2M}(\sum_{i=1}^{Y} U_i)^2 - \frac{1}{2}\sum_{i=1}^{Y}(U_i)^2 \right)$$

with derivatives

$$\frac{\partial L}{\partial U_i} = \begin{cases} Y\mu - M\lambda, & \text{if } i = Y+1 \quad (13a) \\ 1 - \mu - \lambda(2 - U_i - \frac{1}{M}\sum_{i=1}^{Y} U_i), & \text{otherwise.} \quad (13b) \end{cases}$$

A necessary condition for the minimum is that the two derivatives of (13a) and (13b) are zero. One can reformulate each derivative in Eq. (13b) equal to zero such that $U_i$ is a function of $M$, $\mu$, $\lambda$, and $\sum_{i=1}^{Y} U_i$. In doing so, we can notice that all the $U_i$ have the same value. It follows that for all $i = 1, 2, \ldots, Y$:

$$U_1 = U_2 = ... = U_Y \tag{14}$$

To solve these equations, we look at several cases:

**Case 1:** $\mu = 0$. Since the derivative of (13a) must be 0 and $M > 0$, we then have $\lambda = 0$. This in turn results in an infeasible solution in (13b).

**Case 2:** $\lambda = 0$. Similarly, we get $\mu = 0$ by (13a). This also leads to an infeasible solution in (13b).

**Case 3:** $\mu \neq 0$ and, $\lambda \neq 0$. In this case, both constraints (12b) and (12c) must be active. By Eq. (14) and the activation of Eq. (12c), it follows that $U_{Y+1} = U_i$. Therefore, we have $U_1 = U_2 = ... = U_Y = U_{Y+1}$. After solving quadratic equation in one variable with (active) Eq. (12b), we have

$$\forall 1 \leq i \leq Y+1, \quad U_i = \frac{1 + 2\frac{Y}{M} - \sqrt{1 + 2\frac{Y}{M} + 2\frac{Y^2}{M^2}}}{(1 + \frac{Y}{M})} \cdot \frac{M}{Y}$$

It follows that

$$\sum_{i=1}^{Y} U_i = \frac{1 + 2\frac{Y}{M} - \sqrt{1 + 2\frac{Y}{M} + 2\frac{Y^2}{M^2}}}{(1 + \frac{Y}{M})} \cdot M$$

which is identical to Eq. (10). Hence, this theorem is proven. □

Thus, we see that Lemma 2 is necessary for an RAD algorithm to fail; equivalently, the negation of Lemma 2 is a sufficient condition for a system to be feasible for an RAD algorithm using QB:

THEOREM 3. *An implicit-deadline multi-mode system $\tau$ is feasible under RM scheduling and a reasonable allocation decreasing (RAD) algorithm using quadratic bound (QB) if*

$$U^{sum} \leq \left( \frac{3 - \sqrt{5}}{2} \right) M \cong 0.381 \cdot M \tag{15}$$

PROOF. Notice that the right-hand side of Eq. (10) monotonically increases with respect to the value of $\gamma = \frac{Y}{M}$. Thus, the minimization in the right-hand side of Eq. (10) happens when $\gamma$ is minimized. It is evident that at least $M$ tasks can be feasibly assigned to the given processors before the RAD fails; hence, $\gamma \geq 1$. It follows that

$$\frac{1 + 2\gamma - \sqrt{1 + 2\gamma + 2\gamma^2}}{(1+\gamma)} M \geq (\frac{3 - \sqrt{5}}{2}) M \cong 0.381 \cdot M \tag{16}$$

Taking the negation of Lemma 2, this theorem is proven by contrapositive. □

Figure 1 shows the utilization bound under RAD with respect to $\gamma$, assumed to be given. Therefore, if a tighter lower bound on $\gamma$ could be derived, we would be able to achieve better utilization bounds. For example, as shown in Figure 1, if $\gamma \geq 2$, we can conclude a utilization bound of $0.464 \cdot M$.

# 5. BOUNDS BASED ON MAX UTILIZATION

In this section, we derive a generalized utilization bound by considering the upper bound on the utilization for ever tasks $\alpha$. Note that $U_i \leq \alpha$ for every task $\tau_i$. This is motivated by Figure 1. If $\alpha$ is small enough, then, we would like to show that $\gamma$ is also big enough in Lemma 2. We here introduce a new function $\phi(\alpha)$ defined as *the maximum number of tasks with utilization no more than $\alpha$ that can be always guaranteed to fit into one processor.* If we compute $\phi(\alpha)$ by simply combining the total utilization bound of

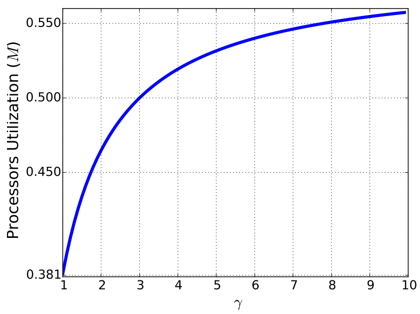


**Figure 1: Utilization bound of RAD with respect to $\gamma$**

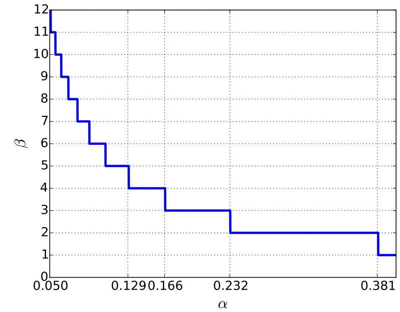


**Figure 2: The value of $\beta$ with respect to given $\alpha$.**

Eq. (1) and the fact that a uniprocessor system with one task is schedulable, we can already obtain a bound on $\phi(\alpha) \geq max\left(\left\lfloor \frac{2-\sqrt{2}}{\alpha} \right\rfloor, 1\right)$. However, such a lower bound on $\phi(\alpha)$ is pessimistic, since the bound $2-\sqrt{2}$ by Eq. (1) comes from a large amount of tasks in [9]. Hence, instead, we consider the quadratic bound, provided in Theorem 1, in hopes of getting a tighter lower bound on $\phi(\alpha)$.

Lemma 3. *The maximum number of tasks with utilization no more than $\alpha$ that can be always guaranteed to fit into one processor $\phi(\alpha)$ is at least:*

$$\phi(\alpha) \geq \beta = \left\lfloor \frac{4+\alpha-\sqrt{\alpha^2+8}}{2\alpha} \right\rfloor \tag{17}$$

Proof. It is not difficult to see the minimum $\phi(\alpha)$ happens when all the tasks are with utilization $\alpha$. The value of $\beta$ can be solved with the following inequality with respect to the upper bound on the utilization for ever task $\alpha$:

$$\alpha \leq 1 - 2\sum_{i=1}^{(\beta-1)} \alpha + \frac{1}{2}\left(\sum_{i=1}^{(\beta-1)} \alpha\right)^2 + \frac{1}{2}\sum_{i=1}^{(\beta-1)} \alpha^2 \tag{18}$$

After reformulation, we obtain

$$\alpha^2(\beta-1)^2 + (\beta-1)(\alpha^2-4\alpha) + 2 - 2\alpha \geq 0 \tag{19}$$

Solving it as the standard quadratic inequality w.r.t. $\beta$, we obtain

$$\beta \leq \frac{4+\alpha-\sqrt{\alpha^2+8}}{2\alpha} \text{ or } \beta \geq \frac{4+\alpha+\sqrt{\alpha^2+8}}{2\alpha} \tag{20}$$

Since we are only interested in providing the lower bound on $\phi(\alpha)$, it is sufficient to take the value with the "$-$" sign as our solution of $\beta$. Thus, it indicates a feasible schedule under RM scheduling on one processor if $\beta$ tasks with utilization no more than $\alpha$ are allocated on the processor. Recall that by definition $\beta$ must be an integer. Hence, we have to apply the floor function to round it down thereafter. We here conclude this lemma. □

Figure 2 depicts $\beta$ with respect to the utilization upper bound $\alpha$. This lower bound is tighter than the one by using only Eq.(1). For example, if $\alpha = 0.381$, we can conclude that $\beta = 2$ by Lemma 3, as can be seen in Figure 2; however, $max\left(\left\lfloor \frac{2-\sqrt{2}}{\alpha} \right\rfloor, 1\right) = 1$. We then make use of Lemma 3 to derive the following generalized utilization bound taking the value of $\beta$:

Theorem 4. *For a given $\alpha$, suppose that $\beta$ is from Lemma 3. A multiprocessor system $\tau$ with implicit deadline, multi-mode tasks is feasible under RM scheduling and a reasonable allocation decreasing (RAD) algorithm using quadratic bound (QB) if*

$$U^{sum} \leq \frac{1+2\beta-\sqrt{1+2\beta+2\beta^2}}{(1+\beta)} \cdot M \tag{21}$$

| $\alpha$ | $(0.381, 1]$ | $(0.231, 0.381]$ | $(0.166, 0.231]$ | $(0.129, 0.166]$ |
|---|---|---|---|---|
| $\beta$ | 1 | 2 | 3 | 4 |
| utilization | $0.381M$ | $0.464M$ | $0.500M$ | $0.519M$ |

**Table 1: Utilization bounds of RAD using QB under $\alpha$**

Proof. By Lemma 3, we have $\gamma = \frac{Y}{M} = \frac{\sum_{j=1}^{M} |\Gamma_j|}{M} \geq \frac{M\beta}{M} = \beta$. Hence, $\beta$ is a lower bound on $\gamma$. Replacing $\gamma$ in Lemma 2 by $\beta$, and similarly taking the negation of Lemma 2, we can thus conclude this theorem. □

This utilization bound is a generalized result of Theorem 3. For example, if $\alpha = 0.23$, then $\beta = 3$. By Theorem 4, we then obtain the utilization bound of 50%, shown in Table 1. Moreover, if $\alpha \to 0$, then $\beta \to \infty$ and the utilization bound converges to $2-\sqrt{2}$:

$$\lim_{\beta\to\infty} \frac{2+\frac{1}{\beta}-\sqrt{2+\frac{2}{\beta}+\frac{1}{\beta^2}}}{(1+\frac{1}{\beta})} M = (2-\sqrt{2})M \approx 0.58M$$

# 6. BOUNDS FOR RELATED MODELS

In this section, we intend to answer the utilization bound on multiprocessor systems for those related models: the generalized multiframe (GMF) model, the variable-rate dependent behavior (VRB) model, and the digraph real-time (DRT) model. As we only focus on implicit-deadline multi-mode task systems, we also implicitly assume that these GMF, VRB, and DRT models are also with implicit deadlines. We explicitly show that the studied mode change model is a relaxation of these models under the above assumption.

Lemma 4. *The studied mode change model is a relaxation of the generalized multiframe (GMF) model , the variable-rate dependent behavior (VRB) model, and the digraph real-time (DRT) model if the minimum inter-arrival time to change to the next mode is always equal to the relative deadline of the current task mode.*

Proof. Due to the space limitation, we only provide sketched explanations here. By definition, a generalized multiframe task [2] must execute each frame sequentially and circularly, starting from any of its frames. The studied multi-mode task model can be equivalently obtained by relaxing the constraint of sequential executions for the generalized multiframe task. Likewise, we can reconstruct the structure of job executions from the directed graph, represented by the DRT model [12], into a complete graph with a self-loop for each vertex. Regarding the VRB model [6], the acceleration or deceleration, that determines which modes may be possibly invoked in the next release under the current rotation speed, can be assumed to be an unlimited value. By considering those execution modes invoked on the threshold of rotation speeds, the corresponding mode change model is thereafter obtained. □

We also conclude the utilization bounds for these models:

Corollary 1. *The generalized multiframe (GMF) task, the variable-rate dependent behavior (VRB) task, and the*

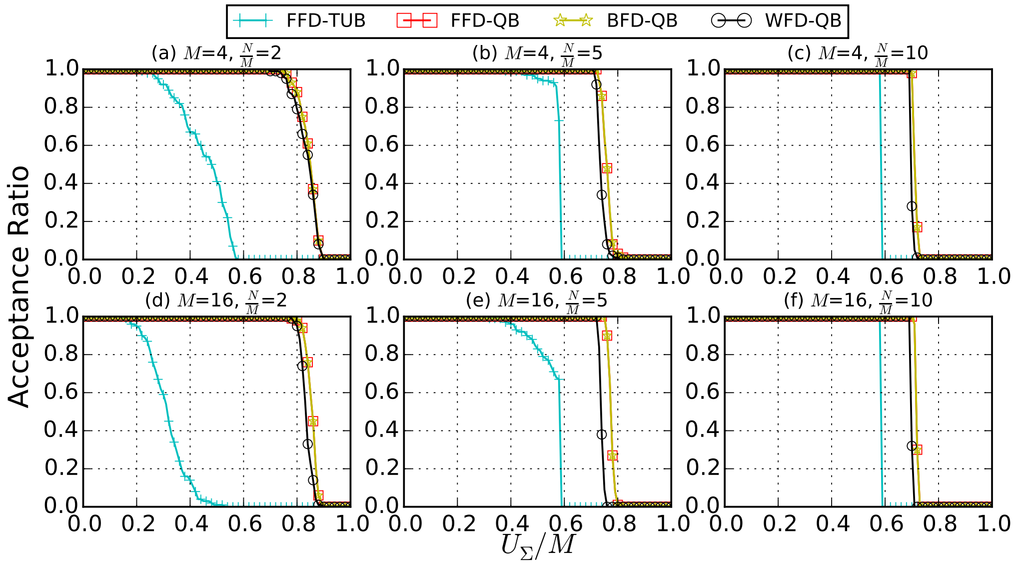


**Figure 3: The acceptance ratio by FFD, BFD, and WFD using the quadratic bound under different numbers of processors and ratios of the number of tasks to the number of processors.**

*digraph real-time (DRT) tasks with implicit deadline on multiprocessor systems achieve the utilization bound provided in Theorems 2, 3, and 4.*

# 7. EVALUATIONS UNDER DIFFERENT RADS

In this section, extensive simulations have been carried out in order to quantify the pessimism of the multiprocessor utilization bounds. The metric to compare the results is to measure the *acceptance ratio* of these tests for a given goal of task set utilization. We generate 100 task sets for each utilization level.

Task utilization values were generated from a uniform distribution but with the constraint that they summed to a constant desired total utilization $U_{\sum}$. We adopted the UUnifast-Discard to generate task sets. The cardinality of a task set was decided according to the ratio of the number of tasks to the number processors, in one of three values $[2, 5, 10]$, provided that the number of available processors is given. We evaluated three RAD algorithms mentioned earlier, namely First-Fit Decreasing, Best-Fit Decreasing, and Worst-Fit Decreasing, using the total utilization bound in Theorem 2, denoted by FFD-TUB, BFD-TUB, and WFD-TUB, receptively, and those using QB in Theorem 3, denoted by FFD-QB, BFD-QB, and WFD-QB, receptively.

The *remaining capacity* $\delta_j$ on processor $j$ for an RAD using TUB and QB is defined as follows: for TUB we use $\delta_j = 2 - \sqrt{2} - \sum_{\tau_i \in \Gamma_j} U_i - U_k$, and for QB we use $\delta_j = 1-2\sum_{\tau_i \in \Gamma_j} U_i + \frac{1}{2}(\sum_{\tau_i \in \Gamma_j} U_i)^2 + \frac{1}{2}\sum_{\tau_i \in \Gamma_j}(U_i)^2 - U_k$, where task $\tau_k$ is the task being allocated.

**Results.** Figure 3 shows the acceptance ratio under difference numbers of processors and tasks, e.g. $M = 16$, $\frac{N}{M} = 5$ implies the number of tasks is $N = 16 \times 5 = 80$. Note that for better readability, we only show the result of FFD-TUB. The performance of FFD-TUB is identical to that of BFD-TUB and slightly better than that of WFD-TUB, similar to the cases where QB is used. The first and most obvious observation is that the RAD algorithm using QB is far more effective than that using TUB. We then notice that FFD-QB and BFD-QB perform alike and are superior to WFD, across all the settings. We also notice that the maximum effective utilization of RADs decreases noticeably from 90% down to 70% when the ratio $\frac{N}{M}$ increases from 2 to 10, with a fixed number of processors (e.g. Figure 3 (a), (b), and (c)). On the other hand, the variance of the number of processors would slightly affect the acceptance ratio, as the ratio of the number of tasks to the number of processors is fixed (e.g. Figure 3 (a) and (d)).

Overall, in our simulations, the simple RAD heuristics combined together with the quadratic bound can admit task sets even with noticeable high utilization, from 70% and up to 90%.

# 8. CONCLUSIONS

A multi-mode task model is a natural generalization model that represents higher expressiveness over the sporadic task model. We study the problem of scheduling multi-mode tasks on multiprocessor systems. We prove that a 38% utilization bound can be guaranteed for implicit-deadline multi-mode tasks on multiprocessor systems by using any reasonable allocation decreasing (RAD) algorithm, e.g. First-Fit Decreasing (FFD), if each mode is prioritized according to rate-monotonic (RM) scheduling policy. Empirical results show that task sets with noticeably high utilization, 70%, are still deemed feasible by our approach.

# References


[1] T. F. Abdelzaher, V. Sharma, and C. Lu. A utilization bound for aperiodic tasks and priority driven scheduling. *Computers, IEEE Transactions on*, 53(3):334–350, 2004.

[2] S. Baruah, D. Chen, S. Gorinsky, and A. Mok. Generalized multiframe tasks. *Real-Time Systems*, 17(1):5–22, 1999.

[3] A. Biondi, M. D. Natale, and G. C. Buttazzo. Response-time analysis for real-time tasks in engine control applications. In *International Conference on Cyber-Physical Systems (ICCPS)*, pages 120–129, 2015.

[4] G. C. Buttazzo, E. Bini, and D. Buttle. Rate-adaptive tasks: Model, analysis, and design issues. In *DATE*, pages 1–6, 2014.

[5] R. I. Davis and A. Burns. A survey of hard real-time scheduling for multiprocessor systems. *ACM Comput. Surv.*, 43(4):35, 2011.

[6] R. I. Davis, T. Feld, V. Pollex, and F. Slomka. Schedulability tests for tasks with variable rate-dependent behaviour under fixed priority scheduling. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2014.

[7] D. de Niz and L. T. X. Phan. Partitioned scheduling of multi-modal mixed-criticality real-time systems on multiprocessor platforms. In *20th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2014, Berlin, Germany, April 15-17, 2014*, pages 111–122, 2014.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[9] W.-H. Huang and J.-J. Chen. Techniques for schedulability analysis in mode change systems under fixed-priority scheduling. In *Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2015.

[10] L. Santinelli, G. Buttazzo, and E. Bini. Multi-moded resource reservations. In *RTAS*, pages 37–46, 2011.

[11] L. Sha, R. Rajkumar, J. Lehoczky, and K. Ramamritham. Mode change protocols for priority-driven preemptive scheduling. *Real-Time Systems*, 1(3):243–264, 1989.

[12] M. Stigge, P. Ekberg, N. Guan, and W. Yi. The digraph real-time task model. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 71–80, 2011.

[13] K. W. Tindell, A. Burns, and A. J. Wellings. Mode changes in priority preemptively scheduled systems. In *Real-Time Systems Symposium*, pages 100–109, 1992.

[14] V. V. Vazirani. *Approximation algorithms*. Springer, 2001.