

两种改进的 EDF 软实时动态调度算法

李 琦¹⁾ 巴 巍²⁾

¹⁾(大连理工大学电子与信息工程学院 辽宁 大连 116024)

²⁾(大连测控技术研究所 辽宁 大连 116013)

摘 要 软实时系统中,任务的随意抢占浪费了 CPU 资源,过多的任务丢失降低了系统的调度性能.文章引入“动态模糊阈值”概念,提出了两种新的类似 EDF 调度的模糊阈值调度算法.两种算法在任务执行过程中,分别通过缩短和延长当前任务截止期至动态模糊阈值的方法节约资源、提高任务完成率.给出模糊阈值的临界值计算公式,分析算法的可调度性,进行仿真与性能分析比较.结果表明,与 EDF 调度算法比较,两种算法有效地提高了重要任务的完成率,当截止期被延长时,任务的错失率明显降低,当截止期被缩短时,任务间的切换次数大大减少.因此,新算法提高了 CPU 的利用率,改善了系统的实时性能.

关键词 EDF;抢占阈值;语言模糊集;模糊规则

中图法分类号 TP316

DOI号: 10.3724/SP.J.1016.2011.00943

Two Improved EDF Dynamic Scheduling Algorithms in Soft Real-Time Systems

LI Qi¹⁾ BA Wei²⁾

¹⁾(School of Electronic and Information, Dalian University of Technology, Dalian, Liaoning 116024)

²⁾(Dalian Scientific Test and Control Technology Institute, Dalian, Liaoning 116013)

Abstract In soft real-time systems, the CPU resource is wasted by switching optionally and the schedulability of the system is decreased by the losing of overmany tasks. The concept of dynamic fuzzy threshold is introduced in this paper, based on which two improved EDF scheduling algorithms are proposed. The deadline of the executing task is delayed to save the resource in one algorithm and is shorten to increase the ratio of success in the other. In this paper, the formulae of the critical value for fuzzy threshold are given and the schedulability is discussed. Finally, the performance of the algorithms is experimentally examined and compared with the EDF algorithm in detail. Results show that, the ratio of success for important tasks is increased efficiently by using these two methods. The missed deadline percentage is decreased obviously while the deadline is delayed and the switching number is reduced greatly while the deadline is shorten. Therefore, by using the new algorithms, the ratio of CPU utilization is enhanced and the real-time capability of the system is improved.

Keywords EDF; preemption threshold; linguistic fuzzy set; fuzzy logic

1 引 言

最小截止期优先(Earliest Deadline First,

EDF)^[1-2]调度算法是实时系统中应用最广泛的调度算法之一,涉及的领域包括工业网络、多媒体传输等众多方面.根据任务的抢占特性不同,EDF 算法分为抢占 EDF 调度算法和非抢占 EDF 调度算法^[3]两

种, 抢占 EDF 调度算法已被证明是最优的动态调度算法^[2], 但是, 调度周期任务和非周期任务, 对于非抢占式调度而言是一个 NP 难题^[3]. 抢占式 EDF 虽然调度灵活、CPU 利用率高, 但内容转换开销远远大于非抢占式 EDF, 对内存的需求也相对较高^[4].

目前, 阈值法和模糊推理法是提升软实时调度性能的两种主要的方法. 文献[4]在文献[5-6]提出的抢占阈值基础上, 针对 ASOS 系统提出了动态抢占阈值调度算法, 把阈值算法引入 EDF 调度算法, 分析任务可调度性能, 给出任务调度方案. 但他仅从整体调度性能上考虑阈值的选取, 只考虑系统完成任务的数量, 而忽略了任务个体之间重要程度的差异, 这可能造成系统中非紧急任务完成出色而重要任务的丢失数量增加, 也将有损系统的调度性能. 文献[7]详细阐述了模糊的理论基础, 为其在调度领域的应用指明了道路. 文献[8]用模糊控制器调度实时任务, 提出最高模糊优先级优先 (Highest Fuzzy Priority First, HFPPF) 调度算法, 它是一种新的调度方案, 很难整合到已广泛使用的 EDF 调度算法中, 且仅针对非抢占系统, 应用范围有所限制.

软实时系统中, 一定数量的任务丢失不会带来致命的错误, 为系统在超载状态下工作提供了便利. 这种情况下, 优化调度性能, 在负载过大时降低对系统资源的需求以及完成更多的任务显得尤为重要. 实际问题中, 任务的重要程度是实时变化的, 它受到多方面的影响, 一个任务在不同的时期具有的重要程度也是不一样的, 这要求任务的优先级考虑多种条件的作用且能够根据需求灵活地变化. 本文提出两种新的基于动态模糊阈值的最小截止期优先调度算法, 由模糊控制器融合多种特性对任务的影响并确定阈值大小, 利用阈值对正在执行的任务进行截止期长短的再调节, 分别在允许范围内虚拟地延长和缩短当前任务的截止期. 仿真试验表明, 两种调度算法下, 重要任务的完成率增加, 利用阈值的方法延长当前任务的截止期, 系统任务的错失率明显减少, 而当前任务的截止期被缩短时, 任务的切换次数显著降低. 新算法降低了 CPU 消耗, 改善了实时调度性能.

2 任务模型

周期任务集 $T = (T_1, T_2, \dots, T_n)$ 中任务 T_i 可以用四元组 (p_i, e_i, D_i, C_i) 来描述, 其中 p_i 表示任务 T_i 的周期; e_i 表示任务 T_i 的执行时间; D_i 表示任务

T_i 的相对时限; C_i 表示任务 T_i 的关键性系数, 取值为 1~7 这 7 个整数, 数值越小任务越关键. 下面给出本文描述任务 T_i 的特性参数.

$T_i(k)$ 表示任务 T_i 的第 k 次执行, $k=1, 2, \dots$.

$d_i(k)$ 表示任务 T_i 的第 k 次执行的截止期

$$d_i(k) = (k-1) \times p_i + D_i.$$

$ret_{ik}(t)$ 表示系统开始 t 时刻后任务 $T_i(k)$ 的剩余执行时间.

$st_{ik}(t)$ 表示系统开始 t 时刻后任务 $T_i(k)$ 的剩余空闲时间

$$st_{ik}(t) = d_i(k) - ret_{ik}(t) - t.$$

$d_i^l(k)$ 表示截止期延长 EDF 算法中任务 T_i 的第 k 次执行时延长后的截止期.

$d_i^s(k)$ 表示截止期缩短 EDF 算法中任务 T_i 的第 k 次执行时缩短后的截止期.

Tr 软实时系统中, 任务的截止期可以一定程度的延长^[9], 表示任务的容忍度为 Tr , 则任务 $T_i(k)$ 的截止期可以延长至

$$d_i^o(k) = (1 + Tr) \times d_i(k), \quad Tr \geq 0 \quad (1)$$

其中 $d_i^o(k)$ 为任务 $T_i(k)$ 的截止期延长上限.

U 表示任务集的处理器的利用率

$$U = \sum_{i=1}^n \frac{e_i}{p_i}.$$

SN 表示任务切换次数, 指从未完成的任务到抢占任务的切换次数.

MDP 表示截止期错失率,

$$MDP = \frac{\text{截止期错失的任务数}}{\text{任务总数}}.$$

3 模糊阈值 EDF 调度算法

3.1 模糊抢占阈值

Wang 和 Saksena 提出了抢占阈值的概念^[6], 扩展了固定优先级的抢占调度. 与此同时, 模糊规则在实时系统中的应用也逐渐兴起^[10-11], 它重新界定了任务调度的界限, 优化了系统的调度性能. 这里我们把模糊阈值的概念引入 EDF 调度算法, 用模糊规则约定阈值系数的大小, 动态控制模糊阈值的变化. 选定任务的剩余空闲时间和关键性指标用语言模糊集表示, 共同作为模糊输入, 其中, 剩余空闲时间标志着任务完成的紧急程度, 它随时间的推移动态变化; 任务的关键性指的是任务在整个实时系统中所处的重要性等级, 它随任务的产生而产生, 且保持不变, 是衡量任务重要程度的标准之一. 在此基础上,

由模糊规则得到该任务的模糊阈值系数,使它的大小和任务自身的特性紧密相关.当前任务将要被新任务抢占时,按照模糊规则计算将要被抢占的任务在当前状态下的阈值系数,并将它的截止期变至它的抢占阈值水平.对于截止期延长的 EDF 算法,任务截止期与它产生的阈值相等直至任务结束,对于截止期缩短的 EDF 算法,如果该任务在缩短截止期后仍被抢占,它将恢复到原截止期,等待下一轮的调度.用模糊阈值优化 EDF 算法的设计图如图 1 所示.

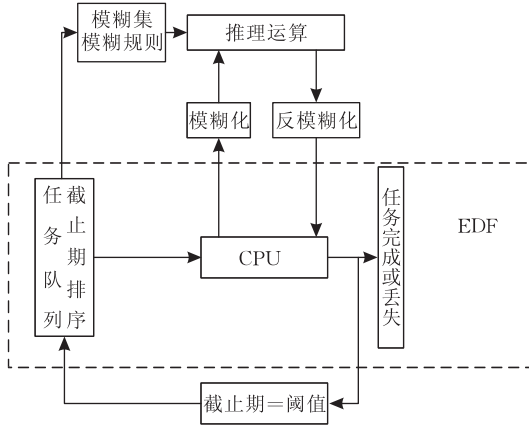


图 1 模糊阈值 EDF 算法设计图

由图 1 可以看出,算法以 EDF 调度算法为基础,增加了模糊推理模块,利用阈值对任务截止期进行调节.采用这种增加模块的形式不仅利于实验室仿真,更为算法软件包的设计与开发提供了方便,加快了其在工业网络式控制系统、现场总线控制系统中应用的步伐.阈值的选取直接影响到算法的性能,本文考虑两种情况:

(1) 利用阈值延长当前任务的截止期.正在执行的任务被抢占后,在容忍范围内延长截止期使得该任务在系统出现空闲后可能继续执行直至完成.

(2) 利用阈值缩短当前任务的截止期.当发生任务抢占时,根据任务的实际情况适当地缩短当前任务的截止期,按照新的截止期重新比较待抢占任务和抢占任务的优先级,使得当前任务有了不被抢占继续执行的可能.

模糊阈值的运用赋予传统的 EDF 调度算法新的活力,它使得任务的抢占特性富于变化,根据任务的紧急情况、关键程度灵活调整任务的抢占特性.当阈值系数为 1 时,这两种调度算法都还原为传统的 EDF 调度算法.

3.2 调度策略

根据截止期的不同变化,提出两种通过动态模糊阈值改进 EDF 算法调度性能的新方案:截止期被

延长的长阈值 EDF(Long Threshold Earliest Deadline First, LTEDF)调度算法和截止期被缩短的短阈值 EDF(Short Threshold Earliest Deadline First, STEDF)调度算法.这里的阈值指的是动态模糊阈值.

3.2.1 LTEDF 调度算法

LTEDF 调度算法利用模糊阈值延长当前任务的截止期.当抢占发生时,计算当前任务 $T_i(k)$ 的阈值系数,设此时 $T_i(k)$ 的阈值系数为 $h_i(k)$ ($h_i(k) \geq 1$),判断 $h_i(k)$ 引起的截止期变化是否超出 $d_i^{to}(k)$ 的范围.如果:

$$h_i(k) \times d_i(k) > d_i^{to}(k) \quad (2)$$

则 $h_i(k)$ 引起的截止期变化已经超出了任务 $T_i(k)$ 容许的范围,此时截止期只能增加到 $d_i^{to}(k)$,即 $d_i^l(k) = d_i^{to}(k)$.如果:

$$h_i(k) \times d_i(k) \leq d_i^{to}(k) \quad (3)$$

则 $h_i(k)$ 引起的截止期变化没有超出任务 $T_i(k)$ 容许的范围,此时任务 $T_i(k)$ 的截止期被延长至 $d_i^l(k) = h_i(k) \times d_i(k)$.

由于截止期被延长,当抢占的任务完成后,如果系统状态空闲,该任务将再次得到执行. $h_i(k)$ 只对任务 $T_i(k)$ 起作用,并不影响 T_i 在其他时段任务的产生以及截止期的变化. $d_i^l(k)$ 将作为 $T_i(k)$ 的截止期参与调度直到任务完成或丢失.此种方案能更最大限度地完成任务、减小错失率且更加有效地利用了系统资源.

3.2.2 STEDF 调度算法

此算法与 LTEDF 调度算法设定的阈值系数相反.在 STEDF 调度算法中,当抢占发生时,首先判断此时当前任务 $T_i(k)$ 的剩余空闲时间 $st_{ik}(t)$ 是否为零,如果 $st_{ik}(t) = 0$,即使再次给该任务分配 CPU 资源,它也将因不满足剩余空闲时间的要求而被丢弃,因此,这种情况无需计算阈值系数,默认 $T_i(k)$ 已丢弃.如果 $st_{ik}(t) > 0$,表示此时当前任务 $T_i(k)$ 的截止期 $d_i(k)$ 仍有可缩短的空间. $d_i(k)$ 缩短的最低下限则由 STEDF 算法产生的阈值和 $st_{ik}(t)$ 共同决定.设该方法下任务 $T_i(k)$ 的阈值系数为 $h'_i(k)$,为了区分任务 $T_i(k)$ 的前后变化,设缩短后的剩余空闲时间为 $st'_{ik}(t)$.为了保证缩短截止期后,任务 $T_i(k)$ 的剩余空闲时间 $st'_{ik}(t)$ 能够满足式(4)的要求, $h'_i(k)$ 在 $h'_i(k) \leq 1$ 的基础上设置下限.

$$st'_{ik}(t) \geq 0 \quad (4)$$

缩短后的截止期表示为

$$d'_i(k) = h'_i(k) \times d_i(k) \quad (5)$$

此时,缩短前的剩余空闲时间表示为

$$st_{ik}(t) = d_i(k) - ret_{ik}(t) - t \quad (6)$$

缩短后的剩余空闲时间表示为

$$st'_{ik}(t) = d'_i(k) - ret_{ik}(t) - t \quad (7)$$

为了满足剩余空闲时间的变化条件,由式(4)和式(7)有

$$d'_i(k) \geq t + ret_{ik}(t) \quad (8)$$

由式(6)有

$$t + ret_{ik}(t) = d_i(k) - st_{ik}(t) \quad (9)$$

把式(5)和式(9)代入式(8)有

$$h'_i(k) \times d_i(k) \geq d_i(k) - st_{ik}(t) \quad (10)$$

由式(10)可得阈值系数 $h'_i(k)$ 变化的下限,即

$$h'_i(k) \geq 1 - \frac{st_{ik}(t)}{d_i(k)} \quad (11)$$

因此 $h'_i(k)$ 的变化范围是

$$1 - \frac{st_{ik}(t)}{d_i(k)} \leq h'_i(k) \leq 1 \quad (12)$$

STEDF 调度算法在设计中,已经要求 $h'_i(k) \leq 1$,为了使 $h'_i(k)$ 在式(12)的范围内变化,计算出阈值系数 $h'_i(k)$ 后,首先要验证 $h'_i(k)$ 是否满足式(11),如满足,则可将 $h'_i(k)$ 代入式(5)计算,如果不满足,则按照式(11),取允许范围内的 $h'_i(k)$ 的最小值 $1 - \frac{st_{ik}(t)}{d_i(k)}$. 在截止期缩短的前提下,STEDF 再次判断该任务是否可以继续得到执行. 如果此时 $d'_i(k)$ 比将要抢占的任务的截止期小,任务 $T_i(k)$ 将继续占用 CPU 资源,本次抢占不发生;如果缩短后的 $d'_i(k)$ 仍然大于将要抢占的任务的截止期,则发生抢占,系统资源让给该抢占任务,任务 $T_i(k)$ 各参数恢复到改变阈值前的状态. 此方案有效地减少了任务

之间的切换,更有利于系统中重要任务的完成.

3.2.3 LTEDF 和 STEDF 算法调度性能和复杂度分析

LTEDF 和 STEDF 调度算法采用模糊阈值系数作为系统对当前任务的截止期的调节手段,使得任务在完全可抢占和不可抢占之间灵活变化. 虽然两种算法的截止期有所改变,但它们仍维持了较好的调度性能.

定理 1. 周期任务集 $T = (T_1, T_2, \dots, T_n)$ 在 $t=0$ 时刻同时释放,若任务集在 EDF 算法中可调度,则在 LTEDF 算法中仍然可以调度.

证明. 在截止期与任务的周期不等的情况下,设周期任务集 $T = (T_1, T_2, \dots, T_n)$ 在 EDF 调度算法中可调度.

则任务集可调度的充分必要条件是^[12-13]

$$U < 1 \text{ 且 } \forall t > 0$$

$$\sum_{i=1}^n \left\lfloor \frac{t + p_i - D_i}{p_i} \right\rfloor e_i \leq t \quad (13)$$

LTEDF 调度算法中,由于任务周期没有发生变化,任务的总利用率不变,任务是否可调度仅取决于是否满足式(13)中 $\sum_{i=1}^n \left\lfloor \frac{t + p_i - D_i}{p_i} \right\rfloor e_i \leq t$ 的要求. 抢占发生时,当前任务 $T_i(k)$ 的截止期增加到了 $d'_i(k)$,设此时相对截止期为 $D'_i(k)$,则相对截止期变化量 ΔD_i 表示为式(14).

$$\Delta D_i = D'_i(k) - D_i = d'_i(k) - d_i(k) > 0 \quad (14)$$

由式(14)有

$$D'_i(k) = D_i + \Delta D_i = D_i + d'_i(k) - d_i(k) \quad (15)$$

则,式(13)的变化可由式(16)所示.

$$\begin{cases} \sum_{j=1}^{i-1} \left\lfloor \frac{t + p_j - D_j}{p_j} \right\rfloor e_j + \sum_{j=i+1}^n \left\lfloor \frac{t + p_j - D_j}{p_j} \right\rfloor e_j + \left\lfloor \frac{t + p_i - D'_i}{p_i} \right\rfloor e_i \leq \sum_{i=1}^n \left\lfloor \frac{t + p_i - D_i}{p_i} \right\rfloor e_i \leq t, & 1 < i < n \\ \left\lfloor \frac{t + p_i - D'_i}{p_i} \right\rfloor e_i + \sum_{j=2}^n \left\lfloor \frac{t + p_j - D_j}{p_j} \right\rfloor e_j \leq \sum_{i=1}^n \left\lfloor \frac{t + p_i - D_i}{p_i} \right\rfloor e_i \leq t, & i = 1 \\ \sum_{j=1}^{n-1} \left\lfloor \frac{t + p_j - D_j}{p_j} \right\rfloor e_j + \left\lfloor \frac{t + p_i - D'_i}{p_i} \right\rfloor e_i \leq \sum_{i=1}^n \left\lfloor \frac{t + p_i - D_i}{p_i} \right\rfloor e_i \leq t, & i = n \end{cases} \quad (16)$$

由式(16)可知,任务集在 LTEDF 调度算法中同样可以调度. 证毕.

定理 2. 按截止期降序排列的周期任务集 $T = (T_1, T_2, \dots, T_n)$ 如果满足

$$\sum_{i=1}^n \frac{e_i}{p_i} \leq 1 \quad (17)$$

$$\forall i, 1 \leq i \leq n; \forall L, p_k \leq L \leq p_i$$

则它在 STEDF 调度算法中可以调度.

证明过程如文献[4]所述.

由定理 1 和定理 2 可知,EDF 可调度是 LTEDF 可调度的充分条件,而 STEDF 的可调度条件与非抢占 EDF 相同.

$$L \geq \sum_{j=1}^{i-1} \left\lfloor \frac{L}{p_j} \right\rfloor e_j + e_i - 1 \quad (18)$$

EDF 算法中由于涉及到对时限的排序,其复杂度为 $O(\log n)^{[14]}$. LTEDF 和 STEDF 调度算法拟采用查表的方式,根据模糊输入查找有限次数的表格即可读取阈值系数,因此,查表部分的算法复杂度为 $O(1)$,总体而言,两种算法的时间复杂度没有改变.

4 性能仿真分析

4.1 阈值系数等级

阈值系数与任务的特性有着密切的关系,它直接反映了该任务此刻在系统中的总体重要程度,本文采用模糊推理求取这两种算法的阈值系数.模糊输入为任务的关键性和剩余空闲时间.这里我们选定最为常用的三角形隶属度函数作为模糊输入和模糊输出的隶属度函数.分别用三个语言模糊集对应不同的输入优先级等级.任务的关键性用“重要”、“一般”、“不重要”来描述,任务的剩余空闲时间用“短”、“中”、“长”描述.得到各自的三角型隶属度函数,如图 2(a)、图 2(b)所示.

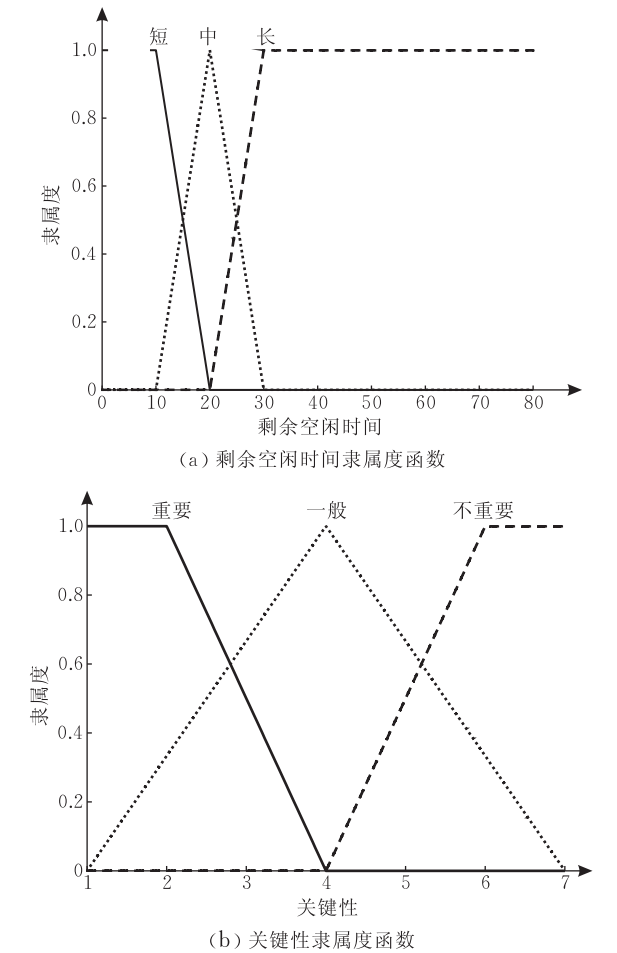


图 2 模糊输入隶属度函数

为了节省系统的运算资源,模糊控制器的输出不再用隶属度函数描述,而是直接用具体的数值表示.这里定义 LTEDF 调度算法的模糊输出为 1、1.25、1.5、1.75、2,STEDF 调度算法的模糊输出为 0.01、0.25、0.5、0.75、1.采用构建模糊逻辑控制表的方法列出各种情况下输入和输出的对应关系,两种方法的控制策略分别如表 1、表 2 所示.发生抢占时,系统当前任务根据其特性得到模糊阈值系数,用模糊阈值代替其截止期参与调度.以 LTEDF 调度算法为例,如果当前时刻正在执行的任务 $T_i(k)$ 的 $st_{ik}(t)$ 为 20, C_i 为 4,则若此时发生抢占,从图 2(a)知 $T_i(k)$ 的剩余空闲时间隶属度为中,从图 2(b)知 $T_i(k)$ 的关键性隶属度为一般,查询表 1,得到它的阈值系数为 1.5. STEDF 模糊逻辑控制表的查询方法与此相同.

表 1 LTEDF 模糊逻辑控制表

C\ST	短	中	长
重要	2.00	1.75	1.50
一般	1.75	1.50	1.25
不重要	1.50	1.25	1.00

表 2 STEDF 模糊逻辑控制表

C\ST	短	中	长
重要	0.01	0.25	0.50
一般	0.25	0.50	0.75
不重要	0.50	0.75	1.00

如表 1 和表 2 所示,第 1 行表示任务空闲时间的等级,第 1 列表示任务关键性程度的等级. LTEDF 和 STEDF 调度算法中,任务按照各自的特点,在表中找到自己的阈值系数.表 1 和表 2 所示的逻辑表中,任务最多查找 3×3 次则可得到阈值系数,因此,查表部分算法的复杂度为 $O(1)$.

4.2 性能仿真

模糊阈值对 EDF 调度算法的性能优化是显著的.为了说明加入模糊阈值后给 EDF 调度算法带来的影响,采用 EDF、LTEDF、STEDF 3 种调度算法进行比较分析.在相同任务模型情况下,分别按 3 个方案得到仿真结果,比较其调度性能之间的差异.

实验采用 MATLAB 软件进行仿真.每次随机产生 5 个任务,记 $U(x,y)$ 表示 x,y 之间随机分布函数,设任务 T_i 的周期 p_i 的取值范围为 $U(5,60)$, C_i 取值范围为 $U(1,7)$, e_i 取值范围为 $U(1,p_i)$,设定初始时刻任务的相对时限 D_i 与任务的周期 p_i 相等,5 个任务在 $t=0$ 时刻同时释放.任务的周期选取的随意性决定了它们的最小公倍数因过大而不适合作

为仿真的持续运行时间,这里限定每次仿真运行的持续时间为 1000 个时间单元,后面所有的仿真结果都是由 100 次独立仿真结果的平均值得到.

在 LTEDF 调度算法中, Tr 决定了阈值的变化空间, Tr 取值不同,系统的性能将产生差异. 选取 Tr 的值为 0.2、0.5 和 1,在相同情况下产生各自的任務错失率,如图 3 所示.

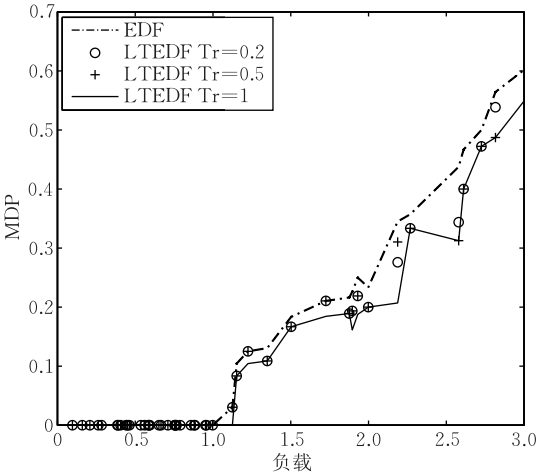


图 3 不同 Tr 下 LTEDF 调度算法错失率对比

从图 3 中可以看出,在负载小于 1 的情况下,各调度算法的错失率均为零,当负载大于 1 时, $Tr=0.2$ 和 $Tr=0.5$ 的 LTEDF 调度算法错失率曲线分别在 $Tr=1$ 的 LTEDF 调度算法和 EDF 调度算法之间. 即 LTEDF 调度算法的任務错失率随着 Tr 的增加而减小,但均小于 EDF 调度算法. 在 $Tr=1$ 情况下, LTEDF 调度算法的错失率最小.

取 $Tr=1$, 分别对 EDF、LTEDF、STEDF 3 种调度算法进行调度性能分析,在相同情况下得到各自的错失率以及切换次数. 错失率对比如图 4 所示,从图中可以看出,3 种算法的 MDP 具有相同的变化趋势,当负载小于 1 时 MDP 几乎为零,当系统过载

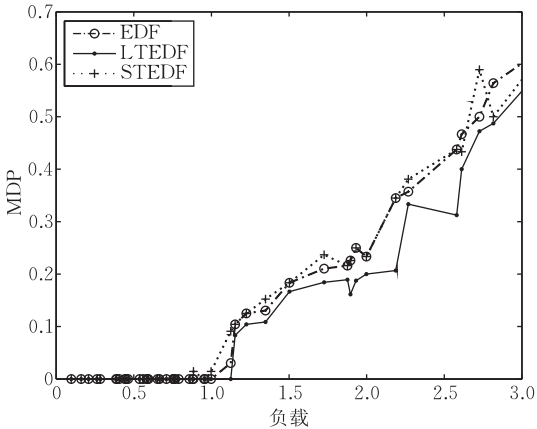


图 4 3 种调度算法的错失率对比

后,MDP 随着负载的增加而增加. 但过载时三种调度算法的错失率各不相同,STEDF 算法的错失率与 EDF 调度算法相差无几,这是因为在 STEDF 算法中,重要的当前任务采用缩短其截止期的方法被率先确保完成,但与此同时,本该抢占当前任务的抢占任务却可能因任务的执行超过截止期而丢失,而 LTEDF 调度算法由于当前任务在抢占任务完成后可能再次执行从而获得了最低的错失率.

3 种调度算法的切换次数对比如图 5 所示. 在相同情况下,3 种算法切换次数变化趋势相同,切换次数在负载为 1 附近时达到最大,从图 5 可知, LTEDF 调度算法的切换次数略小于 EDF 调度算法的切换次数,这是因为 LTEDF 算法中,重要的当前任务在抢占发生时其截止期得到了一定程度的延长,这不仅仅为该任务在抢占任务完成后继续执行直至完成提供了条件,延长的截止期也因相对较小,在一定程度上减少了与后续任务出现任务切换的概率. 而 STEDF 调度算法由于避免了重要的当前任务被抢占,其切换次数明显小于 EDF 调度算法和 LTEDF 调度算法.

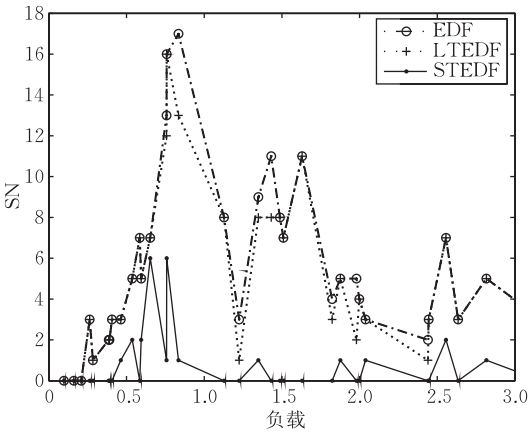


图 5 3 种调度算法的切换次数对比

图 4 和图 5 分别描述了相同情况下任务的错失率和切换次数随负载变化的情况. 从仿真结果图可以看出, LTEDF 和 STEDF 分别改善了 EDF 的 MDP 和 SN 特性,在应用中可以根据实际需要做出选择.

LTEDF 和 STEDF 调度算法采用动态模糊阈值的方法改进 EDF 调度算法的性能,不仅能减少任务的错失率和切换次数,模糊控制器中把任务的剩余空闲时间和关键性作为隶属度,提高了重要任务完成的概率. 图 6 对 3 种调度算法重要任务的完成率进行了比较. 如图 2 所示,选择完成时剩余空闲时间为 $U(0,10)$,关键性为 1、2 的任务为重要任务

进行仿真结果统计,仿真条件同上,从图中可以看出,在负载较轻时,LTEDF 和 STEDF 算法重要任务的完成率与 EDF 调度算法相等,但随着负载的增加,完成率的差距越来越明显,LTEDF、STEDF 调度算法重要任务的完成率普遍高于 EDF 调度算法。

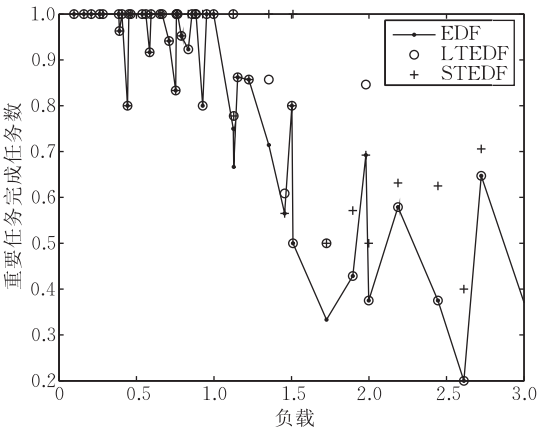


图 6 3 种调度算法重要任务完成情况对比

5 结 论

本文针对软实时系统中 EDF 调度算法存在任务丢失和任务切换的情况,提出两种改进的动态模糊阈值 EDF 调度算法,用模糊逻辑求取抢占阈值系数,把任务本身的特性与阈值的选择紧密关联起来。分别用 LTEDF 减少任务错失率,用 STEDF 降低切换次数,与此同时,利用阈值与任务特性的关系提高重要任务的完成率,仿真实验结果验证了算法的有效性。

LTEDF 和 STEDF 两种算法仅适用于改进软实时系统的抢占调度,且减少任务的错失率和降低任务的切换次数不可兼得,要根据需要优化的参数不同选择不同的算法。动态模糊阈值方法不仅可以用于改进 EDF 调度算法,在实时调度算法尤其是动态调度中,它都能够模糊化调度的标准,提升或降低调度的要求,最终实现优化调度策略的目的。它的通用性将给它带来广阔的应用前景。将动态模糊阈值方法推广到更多的实时调度算法中,通过控制任务的抢占和丢失达到更好的调度效果,是本文下一步的研究工作。

参 考 文 献

[1] Liu C L, Lavland J W. Scheduling algorithm for multiprogramming in a hard real-time environment. *Journal of ACM*, 1973, 20(1): 40-61

[2] Krishna C M, Shin G K. *Real-Time Systems*. Columbus, OH: Mc-Graw-Hill Companies, Inc., 1997: 73-80

[3] Jeffay K, Stanat D F, Martel C U. On non-preemptive scheduling of periodic and sporadic tasks//*Proceedings of the 12th IEEE Symposium on Real-Time Systems*. San Antonio, Texas, 1991: 129-139

[4] He Dong-Zhi, Wang Fei-Yue, Li Wei. Dynamic preemption threshold scheduling for specific real-time control systems//*Proceedings of the 2005 IEEE Networking, Sensing and Control*. Tucson, Arizona, 2005: 395-400

[5] Manas Saksena, Yun Wang. Scalable real-time system design using preemption threshold//*Proceedings of the 21st IEEE RTSS*. Orlando, Florida, USA, 2000: 25-34

[6] Wang Y, Saksena M. Scheduling fixed-priority tasks with preemption threshold//*Proceedings of the 6th International Conference on Real-Time Computer Systems and Application*. Hong Kong, China, 1999: 328-335

[7] Liu Min, Chen De-Gang, Wu Cheng, Li Hong-Xing. Approximation theorem of the fuzzy transform in fuzzy reasoning and its application to the scheduling problem. *Computer and Mathematics with Applications*, 2006, 51(3-4): 515-526

[8] Sabeghi M, Naghibzadeh M, Taghavi T. Scheduling non-preemptive periodic tasks in soft real-time systems using fuzzy inference//*Proceedings of the 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC 2006)*. Gyeongju, Korea, 2006: 27-32

[9] Li Wen-Ming, Kavi Krishna, Akl Robert. A non-preemptive scheduling algorithm for soft real-time systems. *Computer & Electrical Engineering*, 2007, 33(1): 12-29

[10] Kasztenny B, Hatziaadoniu C, Wiszniewski A. Real-time modeling and digital implementation of fuzzy logic controllers//*Proceedings of the 1st International Conference on Digital Power System Simulators (ICDS'95)*. Texas, USA, 1995: 323-328

[11] Sirichuenwicht J, Thumthawatworn T, Santiprabhob P. Adaptive fuzzy rule-based processes scheduling in distributed computing environment//*Proceedings of the 2004 IEEE Aerospace Conference*. Big Sky, MT, USA, 2004, 4: 2249-2262

[12] Baruah S K, Rosier L E, Howell R R. Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. *Real-Time Systems*, 1990, 2(4): 301-324

[13] Hoai Hoang, Buttazzo G, Jonsson M, Karlsson S. Computing the minimum EDF feasible deadline in periodic systems//*Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Application*. Sydney, Australia, 2006: 125-134

[14] Wang Chong-Gang, Long Ke-Ping, Gong Xiang-Yang, Cheng Shi-Duan. The study and perspective of queue scheduling algorithms in packet switching networks. *Acta Electronica Sinica*, 2001, 29(4): 553-559(in Chinese)

(王重钢, 隆克平, 龚向阳, 程时端. 分组交换网络中队列调度算法的研究及其展望. *电子学报*, 2001, 29(4): 553-559)



LI Qi, born in 1979, Ph.D. , lecturer. His research interests include intelligent modeling algorithm, intelligent optimization algorithm, machine learning algorithm, etc.

BA Wei, born in 1980, Ph.D. , engineer. Her research interests include real-time scheduling algorithms and real-time systems.

Background

The research on scheduling algorithms plays an important role in real-time systems. Many new methods were proposed by scholars, which ignore the different importance of the tasks or aim at special cases that are difficult to extend. The performances of switching optionally and losing of over-many tasks weren't improved very well in soft real-time systems. LTEDF and STEDF proposed in this paper assign the priority of the tasks according to their importance flexible using dynamic fuzzy threshold. Comparing to EDF scheduling algorithm, the ratio of success for important tasks is increased efficiently. The missed deadline percentage is decreased obviously in LTEDF scheduling algorithm while the switching number is reduced greatly in STEDF scheduling algorithm. The ratio of CPU utilization is enhanced and the

real-time capability of the system is improved using the new methods. The lab has applied itself to the research of new theory and new algorithms in scheduling technology for a long time. The authors published "A Novel Dynamic Fuzzy Threshold Preemption Scheduling Algorithm for Soft Real-Time Systems" in the proceedings of the 46th IEEE Conference on Decision and Control (CDC), which improves the schedulability of LSF scheduling algorithm using dynamic fuzzy threshold. We focus on the new application of dynamic fuzzy threshold in EDF scheduling algorithm and the schedulability of the algorithms is analyzed carefully in this paper. This makes the method of dynamic fuzzy threshold more extensible and obtains broad application in real-time scheduling area.