Journal of the Korea Convergence Society
Vol. 12. No. 2, pp. 15-20, 2021

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2021.12.2.015

A deferring strategy to improve schedulability for the imprecise convergence on-line tasks

Gi-Hyeon Song
Professor, Division of Medical Business Administration, Daejeon Health Institute Of Technology

부정확한 융복합 온라인 태스크들의 스케쥴가능성을 향상시키기 위한 지연 전략

송기현
대전보건대학교 의료경영과 교수

Abstract The imprecise real-time scheduling can be used for minimizing the bad effects of timing faults by leaving less important tasks unfinished if necessary when a transient overload occured. In the imprecise scheduling, every time-critical task can be logically decomposed into two tasks : a mandatory task and an optional task. Recently, some studies in this field showed good schedulability performance and minimum total error by deferring the optional tasks. But the schedulability performance of the studies can be shown only when the execution time of each optional task was less than or equal to the execution time of its corresponding mandatory task. Therefore, in this paper, a new deferring strategy is proposed under the reverse execution time restriction to the previous studies. Nevertheless, the strategy produces comparable or superior schedulability performance to the previous studies and can minimize the total error also.

Key Words : New deferring strategy, Improve schedulability, Imprecise scheduling, Minimize total error, On-line algorithm, Imprecise convergence on-line task

요 약 부정확한 실시간 스케쥴링은 실시간 시스템에 일시적인 과부하가 발생할 때, 보다 덜 중요한 태스크들을 희생시킴으로서 시간적 오류들의 결과로 발생될 수 있는 나쁜 효과들을 최소화시키기 위하여 사용될 수 있다. 부정확한 실시간 스케쥴링에 있어서, 모든 경성 실시간 태스크는 논리적으로 필수적 태스크와 선택적 태스크로 나누어 질 수 있다. 최근들어서, 선택적 태스크들을 지연시킴으로서 총오류를 최소화시키면서도 필수적 태스크들의 스케쥴가능성을 향상시키는 연구들이 진행되어 왔다. 그러나, 이러한 연구들에서의 스케쥴가능성은 각 각의 선택적 태스크의 실행요구시간이 대응하는 필수적 태스크의 실행요구시간보다 작거나 같을 때 에만 적용된다는 제약이 있었다. 그래서, 본 논문에서는 이전의 연구들의 필수적 및 선택적 태스크들에 대한 실행요구시간 제약조건과 정 반대되는 제약조건 하에서도 적용될 수 있는 새로운 지연 전략을 제시하였다. 그럼에도 불구하고, 본 논문에서 제시한 지연 전략은 총오류를 최소화시키면서도 이전 연구들에 비하여 유사하거나 더 우수한 스케쥴가능성 성능을 보여 주었다.

주제어 : 새로운 지연 전략, 스케쥴가능성 향상, 부정확한 스케쥴링, 총오류의 최소화, 온라인 알고리즘, 부정확한 융복합 온라인 태스크

*Corresponding Author : Gi-Hyeon Song(ghsong@hit.ac.kr)
Received December 4, 2020
Revised January 5, 2021
Accepted February 20, 2021
Published February 28, 2021

1. Introduction

In a (hard) real-time system, every time-critical task must meet its deadline[1]. It is essential for every time-critical real-time task to finish its execution and generate its result by its deadline. Otherwise, a timing fault may occur, and the result produced by the task may be of little or no use[1]. Unfortunately, variations in computing times of dynamic algorithms and excessive crowding on the communication network and so on make meeting all timing constraints at all times arduous[1]. An solution to minimize this trouble is to swap the quality of the results supplied by the tasks with the amounts of processing time required to yield the results[1]. Such a tradeoff can be realized by using the imprecise computation technique[2-4]. In the imprecise scheduling for which the technique can be applied, the problem of scheduling on-line tasks to meet timing constraints and to minimize total error has been studied[5-10]. As a representative example, an algorithm NORA is designed for scheduling on-line task system in which each task is ready upon arrival. The algorithm is optimal in the sense that it can minimize total error under FMC(Feasible Mandatory Constraint)[11]. But, the NORA algorithm did not consider the possibility that the optional tasks can be deferred for helping arrival of the on-line mandatory tasks to meet the FMC[12].

Therefore, [12] proposed an algorithm DOTwP(Deferred Optional Tasks scheduling with Preemption) to improve the schedulability of the oncoming mandatory portions of on-line tasks. The algorithm also keep the total error minimized under the FMC by deferring the optional tasks with preemption. The DOTwP algorithm outperforms 25% ~ 40% better than the NORA algorithm, in terms of the schedulability of the oncoming mandatory portions of on-line tasks[12]. But, the schedulability performance can be shown when the execution time of each optional task was less than or equal to the execution time of its corresponding mandatory task. Thus, in this paper, a new deferring algorithm showing comparable or superior schedulability performance to that of [12] and also keep the total error minimized when the execution time of each optional task is greater than the execution time of its corresponding mandatory task is proposed. The rest of this paper is composed as follows. In section 2, a new optional task deferring algorithm is presented. Next, section 3 shows the simulation study and conclude the paper in section 4.

2. A new optional task deferring algorithm

Let an imprecise task T_i be composed of the mandatory part M_i and the optional part O_i, and characterized by its arrival(ready) time r_i, deadline d_i, and computational requirement m_i and o_i for M_i and O_i, respectively. Let p_i be the sum of m_i and o_i. Assume that at an instant, there are identically arrived N preemptive imprecise convergence on-line tasks running on a single processor and they are sorted according to their deadlines, $d_1 \le d_2 \le \mathrm{K} \le d_{N-1} \le d_N$. First, the algorithm computes the laxity of a task T_i having the fastest deadline. The laxity can be defined as a spare time until before its deadline after scheduling M_i and O_i of the task T_i entirely. If the laxity is greater than 0, then we defer the execution of the optional parts of the tasks having later deadlines than T_i. And instead, the algorithm executes the mandatory parts of the tasks having later deadlines than T_i as much as the laxity until the laxity becomes 0 or closes to 0. But, if the laxity is equal to 0, the tasks M_i

and O_i can be tightly executed in order and else if the laxity is less than 0, some portions of M_i or all portions of M_i and some portions of O_i can be partially executed before its deadline. Whenever some portions of T_i is executed, the laxity is decreased as much as the amount of executed portions of T_i.

Next, the algorithm repeats the process for all imprecise convergence on-line tasks arrived at the instant in ascending order of deadlines. And so, the algorithm can improve the schedulability for mandatory tasks without increasing the total error by executing all portions of T_i and the mandatory parts of the tasks having later deadlines than T_i within the laxity and before its deadline when the task T_i is arrived and scheduled. The following Fig. 1 shows the algorithm.

Theorem.

The DOTBD algorithm is optimal in getting a schedule to minimize total error for an imprecise real-time task system.

Proof.

The optimality of the DOTBD algorithm needs that the following two conditions are satiated :

1. All mandatory tasks are finished by their deadlines.
2. The total error is minimized.

The algorithm computes the laxity of a task T_i whenever the task is arrived and scheduled. When the laxity is greater than or equal to 0, the task M_i and O_i making up T_i can be scheduled entirely before their deadlines. But, if the laxity is less than 0, the algorithm executes only some or all portions of M_i before its deadline prior to O_i as much as the laxity. Therefore, the satisfaction of the condition 1 for any task system is obvious.

Meanwhile, the algorithm executes all portions of T_i and the mandatory parts of tasks having later deadlines than a task T_i until the laxity becomes 0 or closes to 0 whenever the task T_i is arrived and scheduled. By reducing the laxity on and on whenever a task T_i is scheduled, we can improve the schedulability of the mandatory tasks. Moreover, we can minimize the total error by guarantee of the execution for the optional parts of T_i before the mandatoty parts of tasks having later deadlines than T_i. And therefore, the condition 2 is proved. On the other hand, in the proposed DOTBD algorithm showen in Fig. 1, the number of iterations "for loop" is computed can be bounded by $O(N)$, where N is the total number of tasks in the imprecise real-time task system. Next, in the "while loop" of the algorithm, the laxity is decreased more and more as much as the mandatory parts of the tasks having later deadlines than the task T_i when a task T_i is scheduled. So, the number of iterations "while loop" is executed is dependent on the amount of remaining laxity and the unfinished portions of the mandatoty tasks having later deadlines than T_i. Therefore, the number of iterations "while loop" is executed can be bounded by $O(\log N)$. Eventually, the complexity of the algorithm in Fig. 1 becomes $O(N \log N)$.

```
Algorithm DOTBD()
{
  for (i = 1; i <= N; i++)
  {
    task = ListTask[i];
    lxty = d_task − m_task − o_task;
    if (lxty > 0)
    {
      m_task = 0;  o_task = 0;
      j = i + 1;  task = ListTask[j];
      while (lxty >= m_task)
      {
        lxty = lxty − m_task;
        m_task = 0;
        j = j + 1;  task = ListTask[j];
      }
      m_task = m_task − lxty;
      lxty = 0;
    }
    elseif (lxty == 0)  m_task = o_task = 0;
    elseif (lxty < 0)
    {
      if (m_task <= d_task)
      {
        m_task = 0;  o_task = (m_task + o_task) − d_task;
      }
      elseif (m_task > d_task)  m_task = m_task − d_task;
    }
  }
}
```

/* N denotes the number of tasks that can be scheduled in an interval */

/* get a task_id from the task queue ordered by ascending order of task deadlines and computes a laxity of the task corresponding to the task_id */ /* if there is some spare time(laxity) before the task deadline */

/* schedule the mandatory and optional portions of the task in order */

/* while the spare time can be allocated for scheduling the mandatory portions having later deadlines than the current task */

/* schedule the mandatory portions within the laxity */

/* if the remaining laxity is not sufficient for scheduling the mandatory portions then schedule the mandatory potions as much as the remaining laxity. */

/* else if the laxity becomes 0 then schedule the task completely. */ /* else if the laxity is less than 0 and the mandatory portion is less than its deadline then schedule the mandatory portions completely but the optional portion is chopped and partially scheduled

else if the laxity is less than 0 and the mandatory portions are greater than its deadline then the mandatory portions are chopped and partially scheduled. */

Fig. 1. DOTBD algorithm

3. Simulation study

In this section, the process and result of the simulation are described. The aim of the simulation is to compare the schedulability and total error performances of the proposed DOTBD algorithm in Fig. 1 with those of the NORA algorithm[11] and the DOTwP algorithm[12]. In order to estimate the performance of the proposed algorithm, a series of experiments are performed. For each experiment, 100 task sets composed of 100 tasks each, modeled as an M/M/Infinity queuing system, in which the distribution specificity of task reaching time is Poisson, the service time is exponentially distributed are created. The processing time of mandatory part of each task is got uniformly from zero(0) to (its deadline - its ready time) * p, where p is fixed arbitrary less than or equal to 0.4 for each experiment so that the execution time of each optional task is greater than the execution time of its corresponding mandatory task. In each experiment, the schedulability and total error among the proposed DOTBD, the

NORA and the SMF algorithm[13-15] are compared. The following Fig. 2 shows the result of 100 experiments. The simulation result shows that the proposed DOTBD algorithm outperforms the NORA algorithm and is comparable to the SMF algorithm in terms of the schedulability. The SMF(Schedule Mandatory First) algorithm is optimal for schedulability of the mandatory tasks. By the way, the DOTwP algorithm outperforms 25% ~ 40% better than the NORA algorithm, in terms of the schedulability[12].

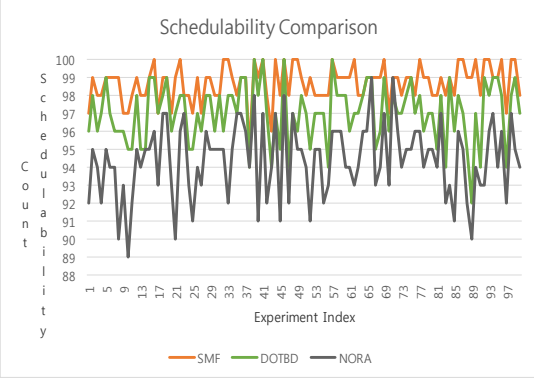

Fig. 2. Schedulability performance comparison

Then, the following Table 1 shows the schedulability performance comparison for the proposed DOTBD algorithm with the DOTwP algorithm. From the Table 1, we can see, the DOTBD algorithm produces better schedulability performance than the DOTwP algorithm in 61 experiments among 100 experiments. And the DOTBD algorithm is comparable to the DOTwP algorithm in 23 experiments among 100 experiments.

But, the DOTBD algorithm produces worse schedulability performance than the DOTwP algorithm in 16 experiments among 100 experiments due to the distribution characteristics of the generated imprecise tasks. Therefore, we can know, the proposed DOTBD algorithm outperforms or comparable to the DOTwP algorithm in the most of experiments from the Table 1.

Table 1. The schedulability performance comparison with the DOTwP algorithm

Improved_Schedulability_Ratio (%) than the NORA	Performance	Count
〉 40	Better than the DOTwP	61
〉= 25 And 〈= 40	Comparable to the DOTwP	23
〉= 0 And 〈 25	Worse than the DOTwP	16
Total		100

4. Conclusion

Recently, the problem of scheduling imprecise convergence on-line tasks to improve the schedulability for the mandatory tasks and to minimize total error for the optional tasks has been studied in some researches to improve the fault-tolerance capability in an (transient) overload situation. The DOTwP algorithm outperforms 25% ~ 40% better than the NORA algorithm for the schedulability of the oncoming mandatory tasks when the execution time of each optional task was less than or equal to the execution time of its corresponding mandatory task.

Meanwhile, to improve the schedulability of the DOTwP algorithm, the DOTBD algorithm is proposed in this paper. The DOTBD algorithm outperforms or comparable to the DOTwP algorithm for the schedulability of the mandatory tasks and keeps the total error minimized when the execution time of each optional task was greater than the execution time of its corresponding mandatory task.

Finally, we will study the effect of the deferring optional tasks on the schedulability and total error depending on the execution time ratioes between the mandatory and optional tasks for further work.

REFERENCES

[1] G. H. Song. (2005). An on-line algorithm to search

minimum total error for imprecise real-time tasks with 0/1 constraint. *Journal of Korea Multimedia Society*, *8(12)*, 1589-1596.

[2] J. Y. Chung, W. K. Shih, J. W. S. Liu & D. W. Gillies. (1989). Scheduling imprecise computations to minimize total error. *Microprocessing and Microprogramming*, *27,* 767-774.

[3] J. W. S. Liu, W. K. Shih, K. J. Lin, R. Bettati & J. Y. Chung. (1994). Imprecise computations. *IEEE Special Issue on Real-Time Systems*, 83-94.

[4] W. K. Shih, J. W. S. Liu & J. Y. Chung. (1991). Fast algorithms for scheduling imprecise computaions. *SIAM Journal on Computing*, *20,* 537-552.

[5] H. P. Choi & Y. S. Kim. (2011). An EDF Based Real-Time Scheduling Algorithm for Imprecise Computation. *Korea Information Processing Society*, *18(4)*, 143-150.

[6] G. H. Song & K. H. Jeon. (2014). A study on the defering method of the optional tasks. *Journal of The Korea Knowledge Information Technology Society(JKKITS)*, *9(1)*, 22-29.

[7] G. H. Song. (2015). A study on new defering method for optional tasks to improve schedulability. *Journal of The Korea Knowledge Information Technology Society(JKKITS)*, *10(3)*, 337~346.

[8] G. H. Song. (2006). An efficient algorithm to minimize total error of the imprecise real time tasks with 0/1 constraint. *Journal of Korea Computer Industry Education Society*, *7(4)*, 309-320.

[9] G. H. Song. (2007). Scheduling algorithm to minimize total error for imprecise on-line tasks. *Journal of Korea Multimedia Society*, *10(12)*, 1741-1751.

[10] G. H. Song. (2007). An improved online algorithm to minimize total error of the imprecise tasks with 0/1 constraint. *Journal of Korean Institute of Information Scientists and Engineers*, *34(10)*, 493-501.

[11] W. K. Shih & J. W. S. Liu. (1996). On-line algorithms for scheduling imprecise computations. *SIAM Journal on Computing*, *25,* 1105-1121.

[12] J. M. Chen, W. C. Lu, W. K. Shih & M. C. Tang. (2009). Imprecise computations with deferred optional tasks. *Journal of Information Science and Engineering*, *25,* 185-200.

[13] S. K. Baruah & M. E. Hickey. (1998). Competitive on-line scheduling of imprecise computations. *IEEE Transactions on Computers*, *47,* 1027-1032.

[14] J. H. Kim, K. Song, K. Choi & G. Jung. (1998). Performance evaluation of on-line scheduling algorithms for imprecise computation. *Proceedings of the 5th IEEE International Conference on Real-Time Computing Systems and Applications*, 217-222.

[15] Riccardo Bettati, Nicholas S. Bowen & J. Y. Chung. (1993). On-line scheduling for checkpointing imprecise computation. *Proceeding of the Fifth Euromicro Workshop on Real-Time Systems*.

송 기 현(Gi-Hyeon Song) [정회원]

· 1985년 2월 : 충남대학교 계산통계학과(이학사)
· 1987년 2월 : 충남대학교 계산통계학과(이학석사)
· 1999년 2월 : 아주대학교 컴퓨터공학과(공학박사)
· 1990년 3월 ~ 현재 : 대전보건대학교 의료경영과 교수
· 관심분야 : 실시간스케쥴링, 레이다 추적, 임베디드시스템, 의료정보시스템, 융복합시스템
· E-Mail : ghsong@hit.ac.kr