Simulation Analysis of Sporadic and Aperiodic Task Handling

Damir Isović and Gerhard Fohler
Department of Computer Engineering
Mälardalen University, Sweden
{dic,gfr}@mdh.se

Abstract

In this paper we present the simulation results for proposed algorithms to handle mixed sets of tasks and sonstraints: periodic tasks with complex and simple constraints, soft and firm aperiodics, and sporadic tasks.

1 Introduction

We have implemented the algorithms described in [2] and [3] and have run simulations for various scenarious.

In the first set of exepriments we simulated the online guarantee algorithm for firm aperiodic tasks, described in [2] (paper B in this thesis). We have studied the guarantee ratio for aperiodic tasks for different combinations of total system loads and aperiodic deadlines.

In the second set of exeperiments we have introduced sporadic tasks, as suggested in [3] (paper C), and have repeated the simulations for different combinations of periodic, sporadic and aperiodic tasks. We have measured the guarantee ratio of firm aperiodic tasks, depending on different scenarious for sporadic tasks. We also investigated how the variations in minimum inter-arrival times for sporadics influence aperiodic guarantee.

The simulation study underlines the effectiveness of the proposed approach.

2 Simulation environment

Simulations refered in previous papers were made with the RTSim simulator [1]. For the purpose of this thesis, we have developed a new simulator to provide for detailed analysis of slot shifting.

We also implemented a debuger, which provides for visual monitoring of the data structures during the simulations.

Simulations were performed in parallell on 5 different PC computers with the processor speed between 333 and 1500 MHz. Some 800 000 different interactions of sporadic, periodic and aperiodic tasks were simulated. The amount of data produced was more than 1 GB. The average size of an input file needed for one graph line was 30 MB. That would produce aproximately 50 MB of data to be analysed. The total lenght of simultions for both experiments was about 200 hours.

3 Experiment 1: Firm aperiodic guarantee

3.1 Experimental setup

For the first experiment series , we randomly generate generated offline and aperiodic task loads, so that the combined load of both periodic and aperiodic tasks was set to 10% - 100%. The deadline for the aperiodic tasks was set to their maximum execution time, MAXT, two times MAXT and three times MAXT. We studied the guarantee ratio of the randomly arriving aperiodic tasks.

The simplest method to handle aperiodic tasks in the presence of periodic tasks is to offline schedule them in background i.e., when there are no periodic instances ready to execute. The major problem with this technique is that, for high periodic loads, the response time of aperiodic requests can be too long. We compared our method to the background scheduling. We refer to our method as *Slot Shifting – Extended*, or SSE.

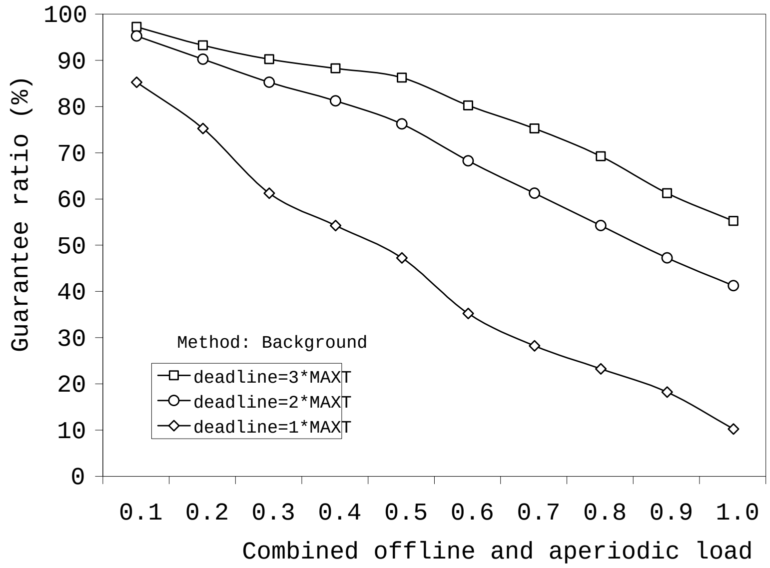

Figure 1. Guarantee ratio for aperiodic tasks – Background

3.2 Results

In this subsection we present obtained results. Each point represents a sample size of 800-3000 simulation runs, with diffrent combinations of periodic and aperiodic tasks. 0.95 confidence intervals were smaller than 5%.

Figure 1 illustrates the performance of background scheduling for three different deadline settings of aperiodic tasks.

Figure 2 depicts the performance of SSE.

In figures 3, 4 and 5 we put both methods together, to see the difference in performance for different deadline settings.

As expected, backgound scheduling performed poorly in the high load situations, specially with tight aperiodic deadlines. For this reason, background scheduling can be adopted only when the aperiodic activities do not have stringent timing constraints and the periodic load is not high.

The graphs show the efficiency of the SSE mechanisms, as guarantee ratios are very high. As expected, the guarantee ratio for aperiodic tasks with larger deadlines is higher than for smaller deadlines. Even under very high load, guarantee ratios stay high.

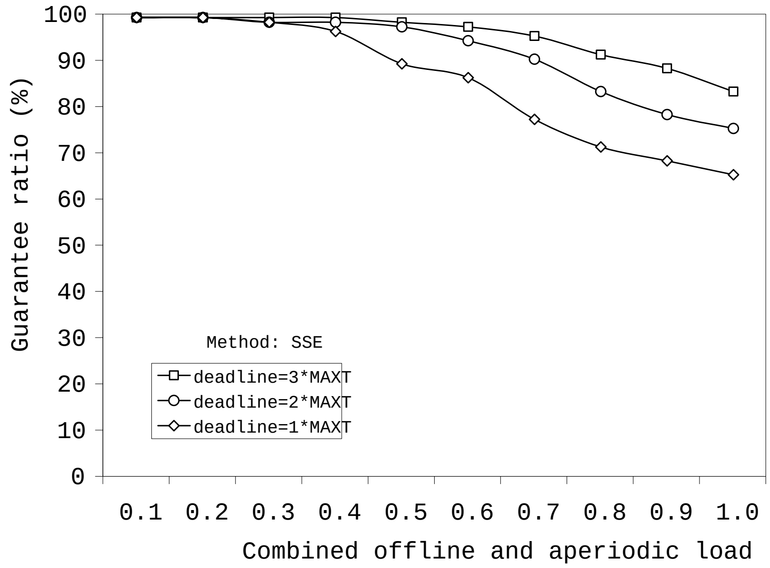

Figure 2. Guarantee ratio for aperiodic tasks – SSE

4 Experiment 2: Firm aperiodic guarantee with sporadics

In the second experiment, we have tested the acceptance ratio for firm aperiodic tasks with the methods to handle sporadic tasks described in in [3]: worst case arrivals without knowledge about invocations (refered as "no info"), and updated worst case with arrival info ("updated").

4.1 Experimental setup

We studied the guarantee ratio of randomly arriving aperiodic tasks under randomly generated arrival patterns for the sporadic tasks. First we investigated the guarantee ratio for firm aperiodic tasks with combined loads 10% - 100%. The deadline for the aperiodic tasks was set to MAXT and 2*MAXT. The combined load was set to 100%.

In the second part of the experiment we varied the arrival frequencies of sporadic tasks according to a factor, f, such that the separation between instances $averageMINT$ is equal to $averageMINT = f * MINT$. This means that if $f = 1$ then the instances are invoked with the maximum frequency, and if $f = 2$, the distance between two consecutive invocations is $2 * MINT$ on average.

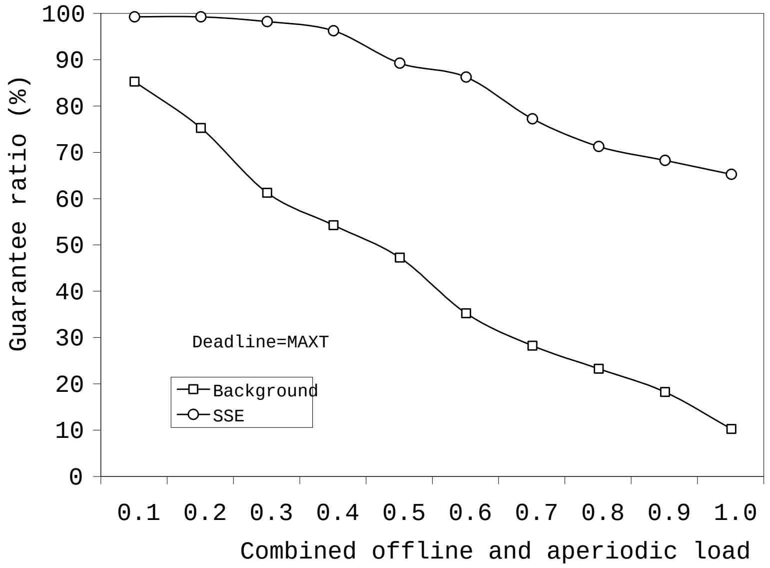

Figure 3. Guarantee ratio for aperiodic tasks, dl=MAXT – SSE vs Background

Figure 4. Guarantee ratio for aperiodic tasks, dl=2*MAXT – SSE vs Bgr

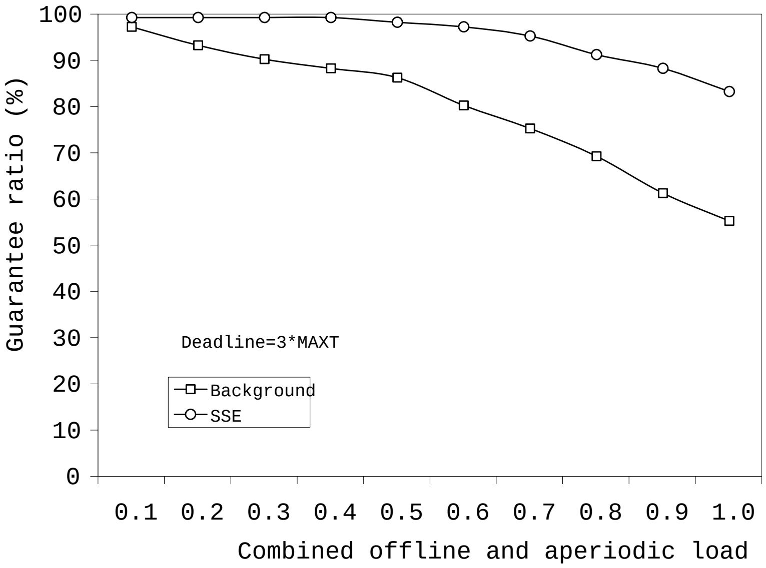

Figure 5. Guarantee ratio for aperiodic tasks, dl=3*MAXT – SSE vs Bgr

4.2 Results

The results from the first part of the experimet experiment are summarized in figures 6 and 7, while the results from the second one are presented in figures 8 and 9.

We can see that our method improves the acceptance ratio of firm aperiodic tasks. This results from the fact that our methods reduce pesimism about sporadic arrivals by keeping track of them.

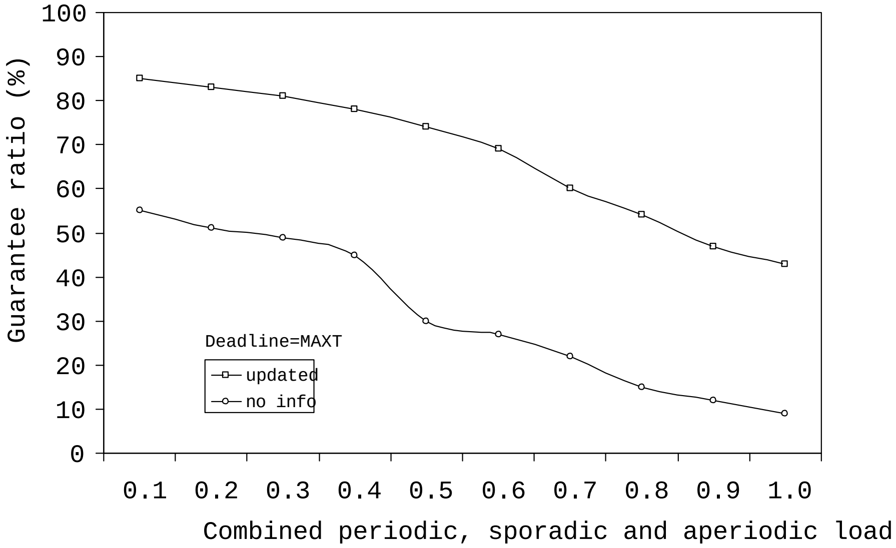

Figure 6. Guarantee ratio for aperiodic tasks with sporadics, dl=MAX: load variation

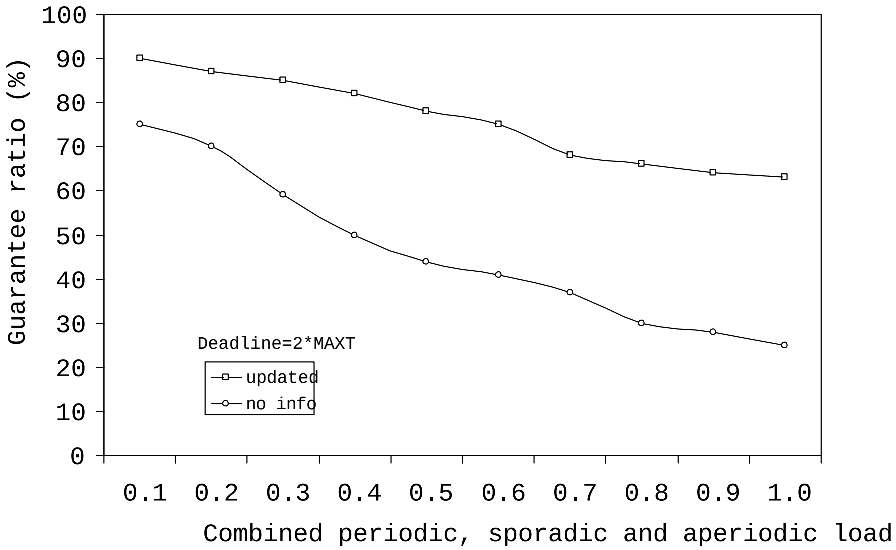

Figure 7. Guarantee ratio for aperiodic tasks with sporadics, dl=2*MAXT: load variation

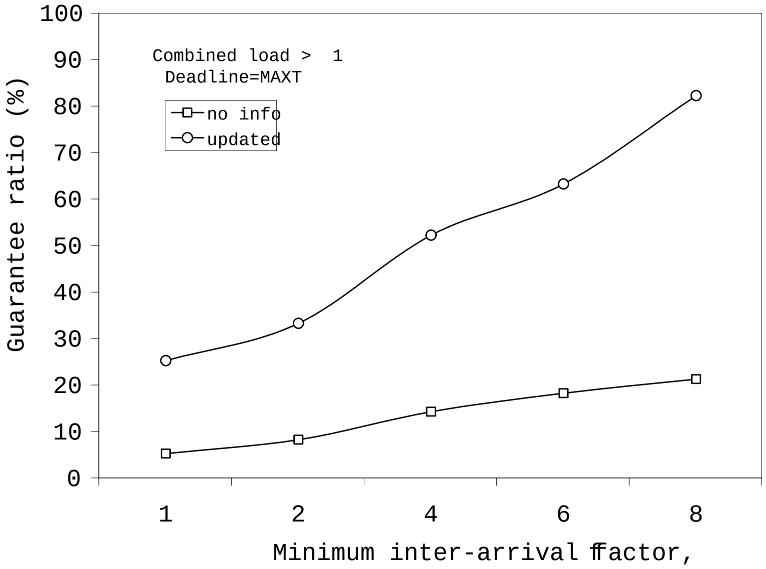

Figure 8. Guarantee ratio for aperiodic tasks with sporadics, dl=MAXT: variation of MINT

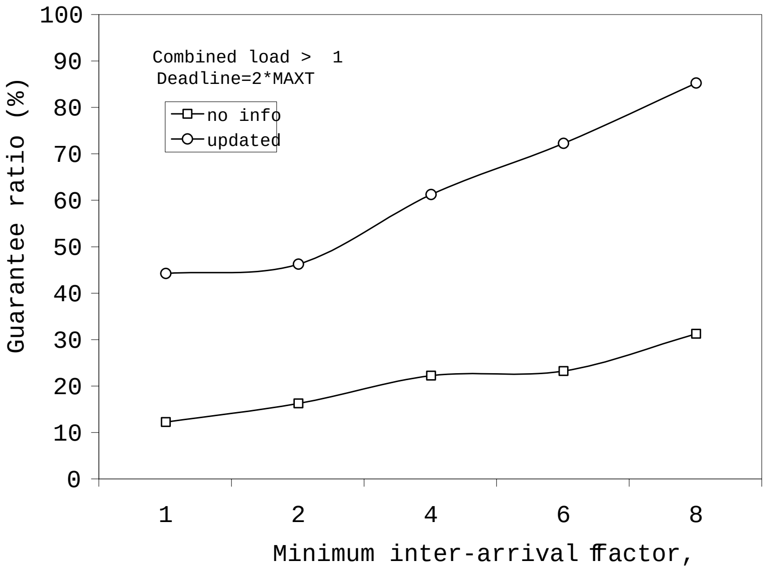

Figure 9. Guarantee ratio for aperiodic tasks with sporadics, dl=2*MAXT: variation of MINT

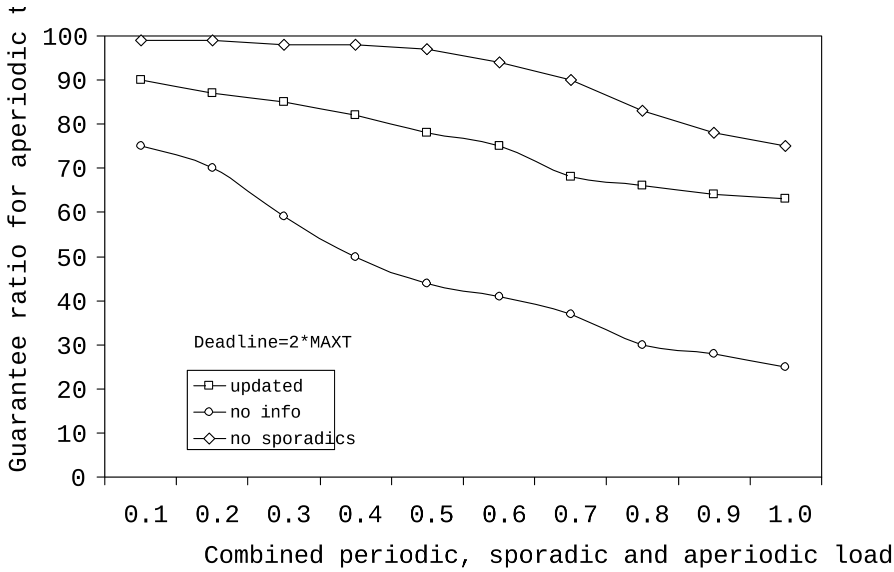

Figure 10. Guarantee ratio for aperiodic tasks with sporadics - Final results

5 Summary

In this report we have presented results of our algorithms to schedule sets of mixed types of tasks with complex constraits, by using earliest deadline first scheduling and offline complexity reduction. In particular, we presented an algorithms to efficently handle sporadic tasks in order to increase acceptance ratio for online arriving firm aperiodic tasks. We have simulated the proposed guarantee algorithms and the results underlines the effectiveness of the proposed approach. Figure 10 summarizes the simulation. We can see that guarantee ratio for firm aperiodic tasks is very high, even when we have sporadic tasks in the system. By keeping track off sporadic arrivals, we can accept firm tasks that otherwise would be rejected.

Future research will deal with the algorithm to include interrupts, overload handling, and complex constraints for both firm aperiodic and sporadic tasks.

References

[1] G. Buttazzo, A. Casile, G. Lamastra, and G. Lipari. A scheduling simulator for real-time distributed systems. In *Proceedings of the IFAC Workshop on Distributed Computer Control Systems (DCCS '98)*, 1999.

[2] D. Isovic and G. Fohler. Online handling of hard aperiodic tasks in time triggered systems. In *Proceedings of the 11th Euromicro Conference on Real-Time Systems*, June 1999.

[3] D. Isovic and G. Fohler. Efficient scheduling of sporadic, aperiodic, and periodic tasks with complex constraints. *In Proceedings of the 21st IEEE Real-Time Systems Symposium, Walt Disney World, Orlando, Florida, USA*, November 2000.