

Evaluation of the Proposed Hybrid Multiprocessor Real-Time Scheduling Approach with Partitioned and Global Approaches

Habibah Ismail^{1,*}, Dayang N. A. Jawawi¹, Ismail Ahmedy², and Mohd Adham Isa¹

¹ Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

² Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

Abstract. Generally, multiprocessor real-time scheduling algorithm fall into two basic approaches, partitioned and global. The hybrid solution that we proposed applies the partitioned scheduling approach to the task set until all processors have been filled. The remaining tasks are then scheduled using the global scheduling approach. The idea of a hybrid scheduling approach to ameliorate limitations of partitioned and global approaches. Studies have shown that most prior research on hybrid multiprocessor real-time scheduling has been confined to hard and soft real-time tasks. In fact, the implementation of hybrid approach and the performance of such algorithms in comparison to partitioned and global approaches have not been fully answered by previous studies. This paper performs experimental evaluation of our proposed hybrid multiprocessor scheduling approach, R-BOUND-MP-NFRNS and RM-US ($m/3m-2$) with multiprocessor response time test, with one of the best scheduling approach from partitioned and global scheduling approaches. The evaluation is based on simulation experiments using videophone application and Cruise Control with Collision Avoidance (CCCA) case studies that mixes real-time tasks of different criticality. The experimental results have presented in terms of three metrics or in other words, performance measurement parameters, namely deadline satisfaction ratio, least system utilization and preemption density (overhead ratio). Based on these three metrics, the proposed hybrid scheduling approach achieved higher percentage compare to the other two approaches. This indicates that such a hybrid scheduling approach is a viable alternative to use in multiprocessor systems and for best scheduled constrained deadline periodic tasks where timing constraints known as weakly hard.

1 Introduction

One way to minimizing resource requirements is through the careful management and allocation, in example, scheduling. Such real-time scheduling has traditionally focused upon scheduling analysis on uniprocessor. Due to the complexity and significantly increased functionality in system computation, thus, attention has been given to multiprocessor scheduling.

There are two types of multiprocessor scheduling, *partitioned* and *global* scheduling algorithms [1]. For partitioned scheduling, it start by partitioning tasks among processors and no migration is permitted on different processors. While, for global scheduling, tasks migration on different processors during execution are permitted.

As a compromise that aims to alleviate limitations of partitioned and global scheduling approaches, intermediate hybrid scheduling has been proposed. This approach use the partitioning as the base of the combination while using the global capabilities to complete it and improve system performance. Accordingly, base of the hybrid multiprocessor real-time

scheduling approach is the partitioning approach, in which it has the minimum overhead and whenever needed, the global approach can be used to improve system utilization and to be near optimal via task migration in order to make use of processors empty capacity for execution of tasks.

In this paper, we considered the schedulability of periodic task systems where timing constraints not to be modeled as hard nor soft, but somewhere in between. A periodic task system can be schedule via a partitioned, global or hybrid approach. Regarding schedulability, we considered weakly hard systems wherein bounded deadline tardiness is permissible, rather than that all deadlines be met. However, the distribution of missed and met deadlines for the tasks during a given time period must be specified precisely [2]. The focus of this study to aforementioned approaches in terms of schedulability considering success ratio, optimality and the tasks migration without increasing the overhead.

Note that we consider static priority algorithms in this paper. Static priority scheduling takes attention because it is an attractive option for real-time applications since it can ensure both the predictability of

* Corresponding author: habibahisma@gmail.com

worst case behaviour and high resource utilization. This is because task will miss deadlines in predictable way and the number of preemptions is bounded.

The objective of this paper is to evaluate algorithms for scheduling when not all such restrictions can be satisfied and to determine the loss to timeliness that relaxing the restriction entails, and also the real challenge is to reduce number of migrations for task.

We evaluate the performance of the scheduling algorithms for multiprocessor real-time systems in the context of resource-limited systems where the number of processors is fixed, and use as our performance metrics the deadline satisfaction ratio, least system utilization and preemption density (overhead ratio) when scheduling a population of task sets.

This paper performs experimental evaluations of R-BOUND-MP-NFRNS and RM-US ($m/3m-2$) with multiprocessor response time test, R-BOUND-MP-NFRNS and multiprocessor response time analysis.

This paper is organized as follows. In section 2, the related previous works are given. In section 3, the evaluation of proposed approach is presented. The simulation result is given in section 4. Finally, our work of this paper is summarized and discussed as conclusions in section 5.

2 Related Works

Various approaches for weakly hard real-time scheduling on multiprocessor are proposed. A weakly hard real-time systems on multiprocessor was first considered in [3]. They proposed the classical weakly hard real-time scheduling algorithms, namely Distance Based Priority (DBP) to apply into multiprocessor applications, called Multiprocessor Distance Based Priority (MPDBPs) to guarantee QoS of both hard real-time tasks and multimedia streams even under overload conditions. In fact, the DBP algorithm originally was introduced by [4] on uniprocessor system. Another work is done by [5] wherein they design new dynamic scheduling algorithm known as the Guaranteed Multiprocessor Real-Time Scheduling (GMRTS-MK) algorithm for (m,k) -firm constrained tasks on homogenous multiprocessors. Later, the same authors, propose an Energy-constrained Multiprocessor Real-Time Scheduling (EMRTS-MK) algorithms for weakly hard real-time for (m,k) -firm deadline constrained tasks running on multiprocessor [6]. Afterwards, in [7] they developed the MC-Fluid scheduling algorithm that executes a task with criticality-dependent rates. The goal is to improve the schedulability of Mixed-Criticality (MC) scheduling algorithms. Practically, not much attention has been devoted to multiprocessor weakly hard real-time scheduling. To the best of our knowledge, we are unaware of any previous study on hybrid scheduling of weakly hard real-time systems.

There are some related works in the area of real-time scheduling, including mixed-criticality systems [8-10]. Also, there exist some works on hybrid scheduling with protocols or artificial algorithms [11-14]. None of these works concentrate on the performance of real-time

systems with aim to be optimal as well as have low overhead besides to guarantee the satisfaction of the timing constraints and parameter of real-time systems.

In [15], hybrid approach is considered in the context of hard and soft real-time systems and the presented evaluation is based on LITMUS^{RT} simulator.

3 Evaluation of Proposed Approach

Previously, in [16] we proposed a hybrid multiprocessor real-time scheduling approach which combines the elements (benefits and advantages) of the two existing approaches. This new hybrid approach was proposed in order to handle the deficits of partitioned and global scheduling approach.

Here, in the evaluation of the proposed scheduling approach (R-BOUND-MP-NFRNS and RM-US ($m/3m-2$) with multiprocessor response time test) with R-BOUND-MP-NFRNS of partitioned and multiprocessor response time analysis (RTA) of global scheduling approaches, necessary metrics are selected and also, validation with Matlab simulation tool with TORSCHE scheduling toolbox for result analysis is performed. This tool suites with scheduling analysis environment and the toolbox covers real-time scheduling [17].

To quantitatively evaluate the system performance of our proposed approach, three measurement parameters of multiprocessor scheduling are used as measurement; *deadline satisfaction ratio*, *least system utilization* and *preemption density (overhead ratio)*. The deadline satisfaction ratio is considered as the primary metric or parameter for evaluation. This is because that metric a recognized performance metric in real-time measures the probability that a task set is schedulable with respect to certain algorithm.

3.1 Deadline Satisfaction Ratio (DSR)

For each experiment, the schedulability of the scheduling approach can be estimated by the term of deadline satisfaction ratio (DSR), which is defined as follows [3, 5, 6]:

$$DSR = \frac{\text{The number of schedulable task sets}}{\text{The number of total task sets}} \times 100\% \quad (1)$$

The deadline satisfaction ratio or success ratio is defined as the number of tasks successfully completed their execution before deadline under a given scheduling algorithm to the total number of tasks admitted in the system for execution. The greater value of success ratio or guarantee ratio has the better performance of schedulability the approach is.

3.2 Least System Utilization (LSU)

Second performance metric is used for evaluating the simulation experiments, namely least system utilization (LSU). The least system utilization is defined as

minimum of the system utilization of all task sets that simulated and found to be unschedulable [18].

$$LSU = \frac{\text{The minimum number of system utilization}}{\text{The number of schedulable task sets}} \quad (2)$$

3.3 Preemption Density

The overhead to resume a pre-empted task on the same processor is the same as when the task resumes on another processor. Hence, the number of pre-emptions during an interval of least common multiple (LCM) can be counted and used as a measure of the performance evaluation.

Note, this gives an indication on the amount of overhead introduced. Preemption density of a task set as a measure of the overhead is defined as the number of pre-emptions during a lcm divided by the lcm itself [5, 18].

$$\text{Preemption Density} = \frac{\text{The number of preemptions during a lcm}}{\text{The least common multiple}} \quad (3)$$

For multiprocessor system, the scheduler runs in one of the processor for assigning the tasks in processors using given algorithm in schedulability analysis.

4 Simulation Result

A performance evaluation of the proposed hybrid, named R-BOUND-MP-NFRNS and RM-US ($m/3m-2$) with multiprocessor response time test with R-BOUND-MP-NFRNS and multiprocessor RTA approaches was conducted. The evaluation is based on simulation experiments using videophone application and Cruise Control with Collision Avoidance (CCCA) task sets. The performance evaluation is conducted using the following experimental setup.

Task sets are generated from the selected case studies and scheduling is simulated with the respective approach on two processors. The execution time, C_i of a task τ_i is computed from the generated utilization of the task. Each task's execution time was determined to be shorter than its period. The total utilization was varied within the range of $(0, 2.0)$, which covers underload and overload situations. All tasks are ordered by increasing their periods, T_i and task τ_1 has the shortest period. The input to the approach parameters is the set of tasks with parameters as mentioned in the task model. The period of a task is generated within the range 50 to 100, to maintain the hyperperiod for the given task set within a limit.

The experimental setting including (n, m) -weakly hard constraints. It provides scheduling optimality for $(1, 1)$ -constrained tasks. For each experimental result, the same experiments were repeated 10 times to obtain the average value.

4.1. Videophone Application

In Fig. 1 until 4, we show timing diagram of each of scheduling approach for videophone application case study.

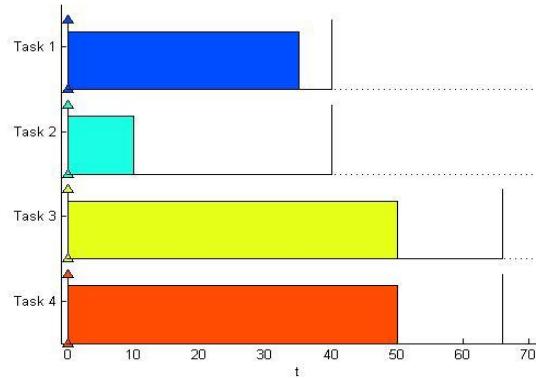


Fig. 1. Each task for videophone application case study.

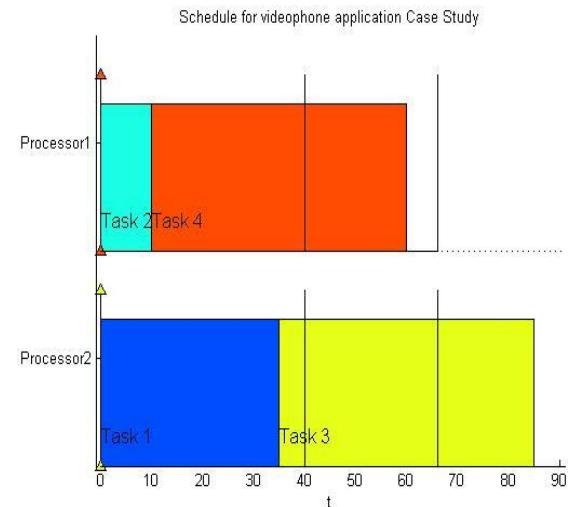


Fig. 2. Timing diagram for videophone application case study using the R-BOUND-MP-NFRNS.

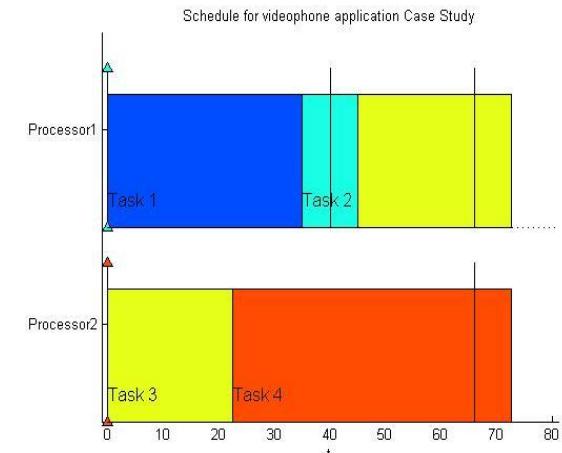


Fig. 3. Timing diagram for videophone application case study using the multiprocessor response time analysis.

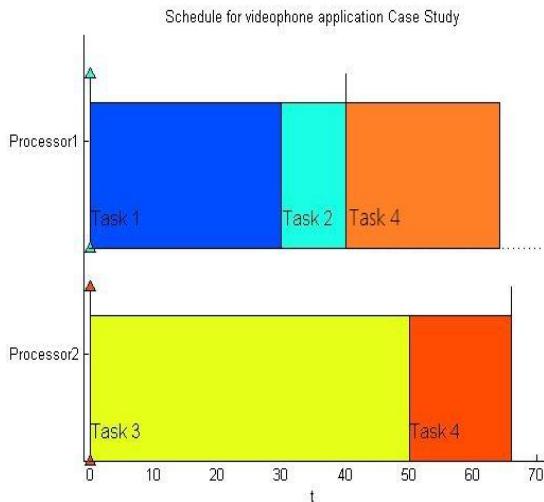


Fig. 4. Timing diagram for videophone application case study using the proposed hybrid scheduling approach.

The experimental results are collected and Table 1 gives the result of simulation for task sets with the utilization within (0.1, 0.5). From the table, it can be seen that our proposed hybrid scheduling approach has a higher satisfaction ratio about 95% than the R-BOUND-MP-NFRNS and multiprocessor response time analysis about 92.5% in a scale of system utilization. The percentage is obtained from simulation tests using three different approaches with same case study, named videophone application. This percentage is based on the number of count task miss of each approach.

From the table, we can also make the following observation. The least system utilization and the deadline satisfaction ratio are correlated. The minimum least system utilization will provide a good or best success ratio.

On average, the preemption density for the proposed scheduling approach is lower than others. The reason for this is that, in hybrid approach, the priorities of the hybrid tasks are always set to be lower than the priority of any partitioned task. With the global approaches, a higher priority task can sometimes execute on another idle processor, thereby avoiding a preemption. For the partitioned approach, an arriving higher priority task must pre-empt a lower priority task.

Another observation is the hybrid scheduling approach consistently outperforms the corresponding approaches and obtained even better performance for weakly hard real-time tasks. This indicates that such a hybrid scheduling approach is viable alternative to use in multiprocessor systems.

Table 1. Simulation results where processor, $m = 2$ using videophone application case study.

| Scheduling Approaches \ Metrics | Proposed Approach | R-BOUND-MP-NFRNS | Multiprocessor Response Time |
|---------------------------------|-------------------|------------------|------------------------------|
| Deadline Satisfaction Ratio | 95% | 92.5% | 92.5% |
| Average Miss Ratio | 0.1 | 0.2 | 0.2 |
| Count Task Miss | 1 | 2 | 2 |
| Least System Utilization | 0.0267 | 0.0355 | 0.0357 |
| Preemption Density | 0.0067 | 0.0069 | 0.0071 |

4.2 Cruise Control with Collision Avoidance (CCCA)

The timing diagram for each of scheduling approach for CCCA case study as shown in Fig. 5 until 8.

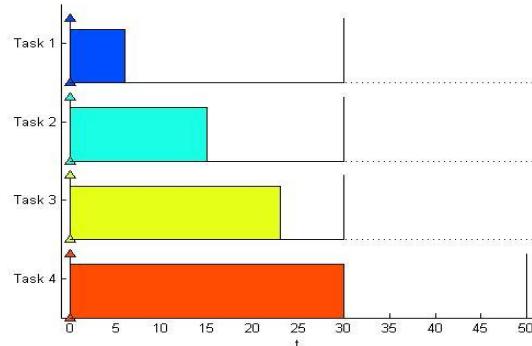


Fig. 5. Each task for CCCA case study.

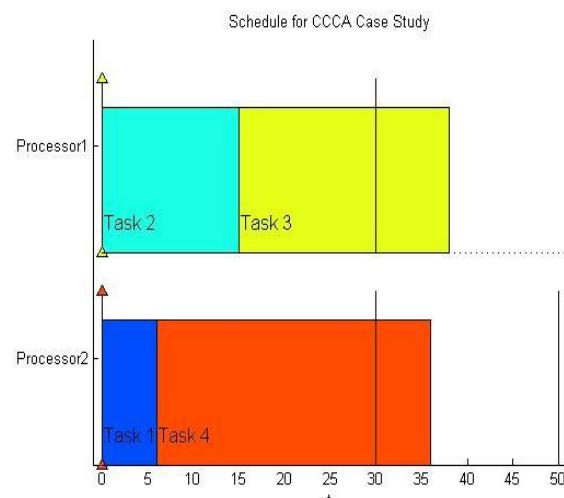


Fig. 6. Timing diagram for CCCA case study using the R-BOUND-MP-NFRNS.

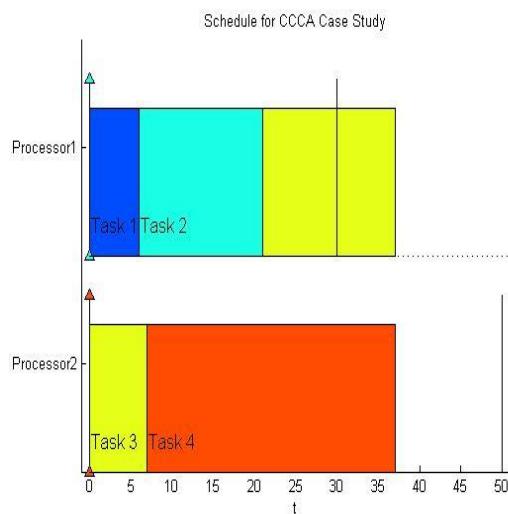


Fig. 7. Timing diagram for CCCA case study using the multiprocessor response time analysis.

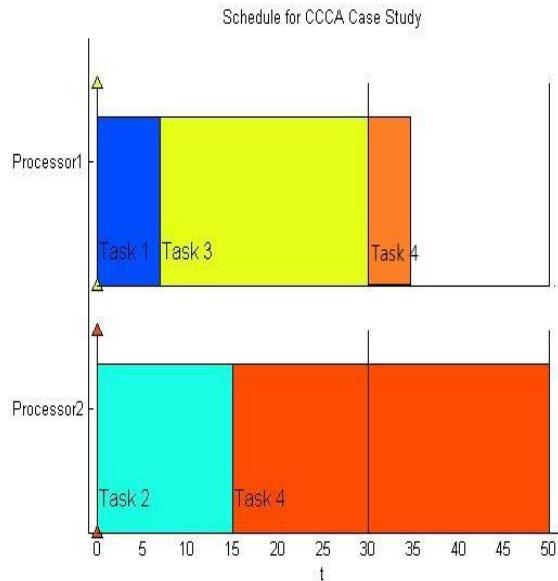


Fig. 8. Timing diagram for CCCA case study using the proposed hybrid scheduling approach.

From the Table 2, it is quite clear to see that our proposed hybrid scheduling approach has a higher satisfaction ratio about 97.5% than the R-BOUND-MP-NFRNS and multiprocessor response time analysis around 95% and 92.5%, respectively. This clearly indicates that our new parametrized priority assignment with Rate Monotonic (RM) scheme works as predicted.

Furthermore, our proposed hybrid scheduling continues to provide a least utilization which is equivalent to that of the best existing partitioning approach, R-BOUND-MP-NFRNS.

Another observation is the number of pre-emptions for our proposed hybrid approach to be less than for the partitioning and global approaches. The simulation results reported in this section indicates that the hybrid scheduling approach in fact perform better than the others.

Table 2. Simulation results where processor, $m = 2$ using CCCA case study.

| Scheduling Approaches \ Metrics | Proposed Approach | R-BOUND-MP-NFRNS | Multiprocessor Response Time |
|---------------------------------|-------------------|------------------|------------------------------|
| Deadline Satisfaction Ratio | 97.5% | 95% | 92.5% |
| Average Miss Ratio | 0.1 | 0.2 | 0.3 |
| Count Task Miss | 1 | 2 | 3 |
| Least System Utilization | 0.0172 | 0.0177 | 0.0179 |
| Preemption Density | 0.0033 | 0.0038 | 0.0040 |

4.3 Discussions

Six experiment simulations are evaluated according to each approach using two case studies. The results are summarize into Table 1 and 2. During simulations, we use five parameters to provide the tests result. The reason of using two different case studies due to compare the result analysis from videophone application's task set with contains more soft functionality, and CCCA task set with mixed task, in which it contains hard and soft functionalities. The result obtained from the both evaluation tables shows that the number of deadlines missed for videophone application is greater than CCCA. This is due to the fact that the task set of videophone application is softer than CCCA in term of the toleration of missed deadline.

The simulation results illustrate that the proposed hybrid multiprocessor real-time scheduling approach (named, R-BOUND-MP-NFRNS and RM-US ($m/3m-2$) with multiprocessor response time test) have even better performance and outperforms approaches that use either partitioned (R-BOUND-MP-NFRNS) or global approach (multiprocessor response time analysis).

Totally, the proposed hybrid multiprocessor real-time scheduling approach performs better than the two other scheduling approaches which are based even on the partitioning or the global multiprocessor real-time scheduling for weakly hard real-time tasks. Our suggested hybrid approach offers one solution to exploiting system resources effectively, while at the same time providing guarantee for those tasks that require so.

5 Conclusion and Future Work

This paper has described the evaluation of our proposed hybrid multiprocessor scheduling approach for weakly hard real-time tasks with partitioned and global approaches. The experimental evaluation was done on these three approaches.

We have evaluated a hybrid solution that combined the overhead characteristics of the partitioned approach with the optimality characteristics of the global approach. The results analysis were investigated through an experiment with the videophone application and CCCA case studies. The performance of the hybrid solution was found to be at least as good as any partitioned and global approaches. Clearly, our scheduling approach can be useful on any multiprocessor system whose worst-case bound delays can be obtained.

In future, this study can be extended by modelling timing constraints and scheduling behaviour through adaptation of modelling language. It can be extended by including a lot more new profiles and compare how far new profiles and schedulability analysis able to work as in interoperability manner.

The authors would like to express profound gratitude to the Universiti Teknologi Malaysia (UTM) and University of Malaya for their financial support under research grant RF006D-2018 and BK043-2015 and not forgotten, the Software Engineering Research Group (SERG) for their help.

References

1. T. P. Baker, S. Baruah, *Handbook of Real-Time and Embedded Systems*, (2007)
2. G. Bernat, A. Burns, A. Llamosi, IEEE TOC, **50(4)**, 308-321, (2001)
3. T. Wu, S. Jin, *1st IEEE International Conference on Ubi-Media Computing*, 320-325, (2008)
4. M. Hamdaoui, P. Ramanathan, IEEE TOC, **44(12)**, 1443-1451, (1995)
5. Y. Kong, H. Cho, 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, 18-23, (2011)
6. Y. Kong, H. Cho, *Computer Science and Convergence*, Lecture Notes in Electrical Engineering 114, 335-347, (2012)
7. J. Lee, *Penn Dissertations*, 1-119, (2017)
8. F. Santy, L. George, P. Thierry, J. Goossens. In *Proceedings of Euromicro Conference on Real-Time Systems (ECRTS)*, 155-165, (2015)
9. T. Fleming, A. Burns, In *Proceedings of Workshop on Mixed Criticality*, IEEE Real-Time Systems Symposium (RTSS), 33-38, (2014)
10. O. Gettings, S. Quinton, R. I. Davis, In *the Proceedings of the 23rd International Conference on Real Time and Networks Systems (RTNS'15)*, 237-246, (2015)
11. M. R. Mohamed, M. H. A. Awadalla, IJCSI, **8(3)**, 79-89, (2011)
12. M. Tamut, K. Chakma, IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 1153-1157, (2014)
13. S. Afshar, M. Behnam, R. J. Bril, T. Nolte, *IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, (2015)
14. H. Kazemi, Z. M. Zahedi, M. Shokouhifar, ECIJ, **5(1)**, 1-13, (2016)
15. A. Bastoni, B. B. Brandenburg, J. H. Anderson, *31st IEEE Real-Time Systems Symposium*, 14-24, (2010)
16. H. Ismail, D. N. A. Jawawi, *Modeling, Design and Simulation of Systems*, 666-678, (2017)
17. M. Kutil, P. Sucha, M. Sojka, Z. Hanzalek, *TORSCHE Scheduling Toolbox for Matlab*, User's Guide, (2010)
18. B. Andersson, J. Jonsson, *Proceedings Seventh International Conference on Real-Time Computing Systems and Applications*, (2000)