International Journal of Students' Research in Technology & Management
eISSN: 2321-2543, Vol 6, No 1, 2018, pp 14-17
https://doi.org/10.18510/ijsrtm.2018.613

GENETIC ALGORITHM WITH TWO OBJECTIVE FOR REAL-TIME TASK SCHEDULING WITH COMMUNICATION TIME

[1*]Myungryun Yoo, [2]Takanori Yokoyama
[1 2]Department of Computer Science, Tokyo City University, Japan
*myoo@tcu.ac.jp, tyoko@tcu.ac.jp

Article History: Received on 17th February, Revised on 01st March, Published on 04th April 2018

Abstract

Purpose of the study: The real-time task scheduling on multiprocessor system is known as an NP-hard problem. This paper proposes a new real-time task scheduling algorithm which considers the communication time between processors and the execution order between tasks.

Methodology: Genetic Algorithm (GA) with Adaptive Weight Approach (AWA) is used in our approach.

Main Findings: Our approach has two objectives. The first objective is to minimize the total amount of deadline-miss. And the second objective is to minimize the total number of processors used.

Applications of this study: For two objectives, the range of each objective is readjusted through Adaptive Weight Approach (AWA) and more useful result is obtained.

Novelty/Originality of this study: This study never been done before. This study also was provided current information about scheduling algorithm and heuristics algorithm.

Keywords: real-time task scheduling algorithm; two objective genetic algorithm; adaptive weight approach; communication time; execution order

INTRODUCTION

Several scheduling algorithms based on Rate Monotonic (RM) and Earliest Deadline First (EDF) are proposed for hard real-time system with uniprocessor (Diaz, Garcia & Lopez, 2004; Bernat, Burns & Liamosi, 2001). Every tasks are finished their execution before their deadline in these algorithms. However, the overloaded situation is not considered in these algorithms. Recently, a lot of soft real time applications are being used and the scheduling algorithm for soft real-time system is needed. In soft real-time system, some deadline-miss is acceptable. Figure 1 represents the definition of deadline-miss.

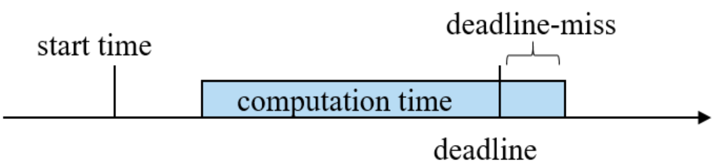

Figure 1: The definition of deadline-miss

Several algorithms for soft real time system are proposed (Kim, Lee & Lee, 1997; Yoo, 2016). However, these algorithms have some improvement in deadline-miss under the overloaded situation. Furthermore, the scheduling problem on multiprocessor is an NP-hard problem (Yalaoui & Chu, 2002). Consequently algorithms based on heuristics have been proposed.

LITERATURE REVIEW

Recently, several scheduling algorithms using Genetic Algorithm (GA) are proposed for single objective (Mitra & Ramanathan, 1993; Lin & Yang, 1999; Monnier, Beauvais & Deplanche, 1998). These algorithms have only one objective such as minimizing cost, finish time, deadline-miss.

Oh and Wu proposed GA for task scheduling on multiprocessors (Oh & Wu, 2004). Their approach has two objectives; to minimize the total amount of deadline-miss and the total number of processors used. However this algorithm did not consider the confliction between objectives, so called Pareto Optimum.

In this paper, we propose a new scheduling algorithm using Genetic Algorithm (GA) for in soft real-time multiprocessor system with the communication time between processors and the execution order between tasks. The objective of proposed approach is to minimize the total amount of deadline-miss and the total number of processors used.

METHODOLOGY

Our scheduling problem is defined as to assign tasks to homogeneous multiprocessors with communication time and the objective of scheduling is to minimize the total number of processors used f_1 and the total amount of deadline-miss f_2. Figure2 represents the definition of our scheduling problem.

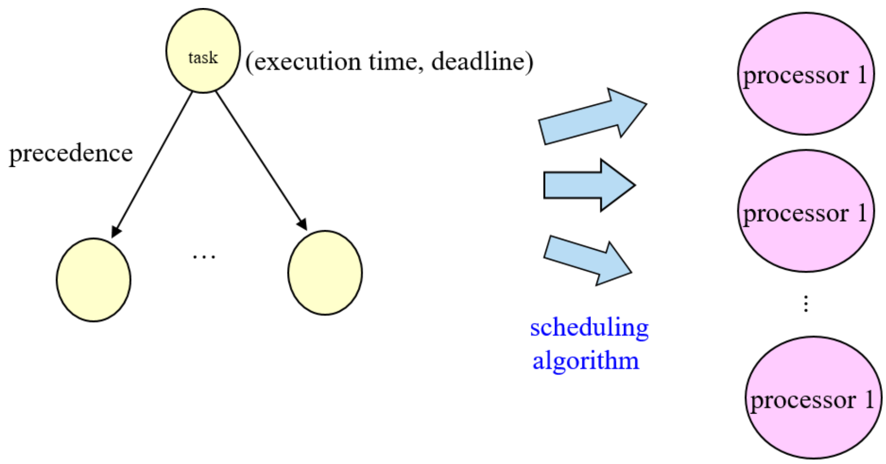

Figure 2: The definition of scheduling problem

The proposed approach is based on genetic algorithm (GA). In our GA, a chromosome V represents the information of task assignment to processor and execution order. A chromosome V is consisted of two rows. The above row V_1 represents execution order and the below row V_2 represents assignment information. The length of a chromosome is the total number of tasks. The execution order should be satisfied a precedence constraints on the given task graph.

Figure 3 represents the example of a chromosome.

total number of tasks

V	$task_1$	$task_2$	$task_3$	$task_4$	…
V_1	2	3	1	4	…
V_2	2	1	3	2	…

Figure 3: The example of a chromosome

In Figure 3, the matrix (2 by the total number of tasks) is basically filled with a random decimal number. The random number of V_1 means the execution order. Firstly, task3 is executed. And then task1 will be started. The third execution task is task2. The random number of V_2 means the assigned processor number. The task1 and task4 are executed on the processor2. The task2 is executed on the processor1. However, the total number of processor is not fixed. It is set by initial value in the step2 of encoding procedure and changed automatically in during of loop (generation of GA). We can obtain the final total number of processor after the step2 of encoding procedure.

For another operation of GA, we use one cut crossover and mutation of only one bit in V_2. And we use roulette wheel selection.

For the evaluation of two objective f_1 and f_2, the following equation is used.

$$eval(V) = \frac{1}{F(V)} = \frac{1}{\sum_{i=1}^{2} \frac{f_i(V)}{f_i^{max} - f_i^{min}}} \quad (1)$$

FINDINGS / RESULTS

We evaluate the performance of proposed algorithm (two-GA) through several simulation. We compared proposed two_GA with Oh-Wu's algorithm by Oh and Wu.

Table 1 shows the comparisons of results of randomly generated task graph with 50 tasks. Tasks' computation time, deadline and communication time between processors are generated randomly based on normal distribution and exponential distribution.

Table 1: Computation results (50 tasks)

Terms	Normal distribution		Exponential distribution	
	Oh-Wu's Algorithm	**Two_GA**	**Oh-Wu's Algorithm**	**two_GA**
Total number of processors	35	32	32	28
Final finish time	47	52	46	52
Computing times (*msec*)	128	132	135	136
Average utilization of processors	0.47568	0.49376	0.43674	0.47253

In the case of no deadline-miss inclusively, the computing time of proposed two_GA is longer than that of Oh-Wu's algorithm. However, the total number of used processors is fewer than that of Oh-Wu's algorithm. Also, the average utilization of processors of our algorithm is higher than that of Oh-Wu's algorithm.

Table 2 shows the comparisons of results of randomly generated task graph with 100 tasks under the same condition of Table 1. In this case, we can see the same results to Table 1.

Table 2: Computation results (100 tasks)

Terms	Normal distribution		Exponential distribution	
	Oh-Wu's Algorithm	**two_GA**	**Oh-Wu's Algorithm**	**two_GA**
Total number of processors	36	34	36	32
Final finish time	162	171	198	199
Computing times (*msec*)	455	498	478	503
Average utilization of processors	0.40582	0.43392	0.43363	0.48823

Figure 4, Figure 5, Figure 6 and Figure 7 represent the Pareto solution of proposed two_GA and that of Oh-Wu's algorithm. In these figures, the Pareto line by proposed two_GA is closer to ideal point (1, 0) than that of Oh-Wu's algorithm.

CONCLUSION

In this paper, we proposed a new task scheduling algorithm using GA. We consider the scheduling problem on soft real-time multiprocessor system with the communication time between processors and the execution order between tasks. The objective of proposed approach is to minimize the total amount of deadline-miss and the total number of processors used. Our GA is combined Adaptive Weight Approach (AWA) to satisfy two objectives simultaneously. From the simulation tests, the performance of the proposed two_GA are better than that of other algorithms.

LIMITATION AND STUDY FORWARD

We plan to design real-time tasks scheduling algorithm on heterogeneous multiprocessors system for the next step of study.

ACKNOWLEDGEMENT

This work supported in part by JSPSKAKENHI Grant Number 15K00084.

REFERENCES

1. Bernat G., Burns, A. & Liamosi, (2001). A. Weakly Hard Real-Time Systems. *Transactions on Computer Systems*, 50(4), 308-321. https://doi.org/10.1109/12.919277
2. Diaz J. L., Garcia, D. F. & Lopez, J. M. (2004). Minimum and Maximum Utilization Bounds for Multiprocessor Rate Monotonic Scheduling. *IEEE Transactions on Parallel and Distributed Systems*,15(7), 642-653. https://doi.org/10.1109/TPDS.2004.25
3. Kim M. H., Lee, H. G. & Lee, J. W. (1997). A Proportional-Share Scheduler for Multimedia Applications. *Proc. of Multimedia Computing and Systems*, 484-491.
4. Lin, M., & Yang, L. (1999). Hybrid Genetic Algorithms for Scheduling Partially Ordered Tasks in A Multi-processor Environment. *In: Proceedings of the 6th International Conference on Real-Time Computer Systems and Applications*, 382–387.

5. Mitra, H., & Ramanathan, P. (1993). A Genetic Approach for Scheduling Non-preemptive Tasks with Precedence and Deadline Constraints. *In: Proceedings of the 26th Hawaii International Conference on System Sciences*, 556–564. https://doi.org/10.1109/HICSS.1993.284070
6. Monnier, Y., Beauvais, J. P. & Deplanche, A. M. (1998). A Genetic Algorithm for Scheduling Tasks in a Real-Time Distributed System. *Proc. of 24th Euromicro Conference*, 708-714. https://doi.org/10.1109/EURMIC.1998.708092
7. Oh, J., & Wu, C. (2004). Genetic-algorithm-based Real-time Task Scheduling with Multiple Goals. *Journal of Systems and Software*, 71(3), 245-258. https://doi.org/10.1016/S0164-1212(02)00147-4
8. Yalaoui, F., & Chu, C. (2002). Parallel Machine Scheduling to Minimize Total Tardiness. *International Journal of Production Economics*, 76(3), 265–279. https://doi.org/10.1016/S0925-5273(01)00175-X
9. Yoo, M. (2016). Continuous Media Tasks Scheduling Algorithm. *International Journal of Electronics Communication and Computer Engineering*, 7(2), 99-103.

