Volume 4, Issue 6, June 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

AI Techniques for Real-Time Systems

Apurva Shah*
Faculty of Technology and Engineering,
M S University of Baroda
India

Abstract—Artificial Intelligence(AI) is an area which grows very fast along with the development of hardware and software technology. AI methods are moving towards more realistic domains requiring real-time responses and real real-time systems are moving towards more complex applications requiring intelligent behaviour. Before few years of era, it was believed that AI cannot be used for real-time systems. But now, it has been tried to identify the various goals of real-time system that can be achieved using vast AI techniques. In this paper, it has been discussed this complex problem and tried to identify promising areas for future research in AI techniques to target real-time challenges.

Keywords— Real-time systems, Artificial Intelligence, Swarm Intelligence, Scheduling

I. INTRODUCTION

Background of real-time systems and AI should be discussed first. Thereafter, application of AI for real-time systems has been discussed.

(A) Real-Time Systems

Real-time systems are defined as those systems in which the correctness of the system depends not only on the logical results of computations but also on the time at which the results are produced [1]. Real-time system is a processing system that is designed to handle workloads whose tasks have completion deadlines. The objective of the real-time system is to meet these deadlines; that is, to process tasks before their deadlines expire. Therefore, in contrast to conventional computer systems where the goal usually is to minimize task response times, the emphasis here is on satisfying the timing constraints of tasks.

Main characteristics of real-time systems are:

- Multi tasking
- Priority scheduling
- Programmer defined interrupts
- Support of real-time clock
- Special alarms and timeouts

Real-time systems play a vital role in our society and the spectrum of complexity of such systems vary widely from the very simple to the extremely complex systems e.g. patient monitoring system in an ICU, flight control system, nuclear power plants, space shuttle, multimedia, robot control in automated factory, weather warning system, etc.

They can be categorized in two basic types: Hard and Soft.

(a) Hard Real-Time Systems are having very hard deadline i.e. absolute deadlines must be met otherwise catastrophic situation may arise. Nuclear power plant and flight control system are example of hard real-time systems.

(b) Soft Real-Time Systems are with soft deadline. It requires that critical tasks get priority over less prior task. In this system, missing an occasional deadline is undesirable but nevertheless tolerable. Multimedia is an example of soft real-time systems.

For hard real-time systems as catastrophic failure may occur if deadlines are missed, control systems for real-time environments must take appropriate actions at appropriate times only. Research in real-time systems addresses precisely this issue, by developing methods for guaranteeing that the reaction rate of a control system matches the rate of change in the environment. Therefore, these systems are required to be predictable.

For soft real-time systems timing constraints don't lead to catastrophic conditions and it gives a minor flexibility to increase the speed at the cost of predictability.

The objective of real-time task scheduler is to guarantee the deadline of tasks in the system as much as possible when we consider soft real-time system. To achieve this goal, vast researches on real-time task scheduling have been conducted. Mostly all the real-time systems in existence use preemption and multitasking. Real-time scheduling techniques can be broadly divided into two categories: Static and Dynamic.

Static algorithm assigns all priorities at design time, and it remains constant for the lifetime of a task. Dynamic algorithm assigns priority at runtime, based on execution parameters of tasks. Dynamic scheduling can be either with static priority or dynamic priority. RM (Rate Monotonic) and DM (Deadline Monotonic) are examples of dynamic scheduling with static priority [2]. EDF (Earliest Deadline First) and LST (Least Slack Time First) are examples of dynamic scheduling with dynamic priority. EDF and LST algorithms are optimal under the condition that the tasks are preemptive, there is only one processor and the processor is not overloaded ([5],[6]). In fact, it has been experimentally demonstrated that these algorithms perform quite poorly when the system is overloaded ([3],[4]).

(B) Artificial Intelligence

AI research attempts to computationally model the various intelligent capabilities of humans as well as nature and, adapting to dynamic and uncertain techniques viewed as search. Traditional AI techniques are now enriched with development of various soft computing techniques and swarm intelligence. Genetic algorithm, fuzzy logic etc soft computing techniques have grown up as along with the technology.

Swarm Intelligence (SI) is computational and behavioural metaphor for the problem solving that originally took its inspiration from the Nature's examples of collective behaviours. SI provides a basis, which makes it possible to explore collective (or distributed) problem solving methodology without centralized control or the provision of a global model [7]. SI makes System-level behaviour appear to transcend the behavioural repertoire of single (minimalist) agent. ACO is the branch of SI. The ACO algorithms are computational models inspired by the collective foraging behaviour of ants [8]. In Ant Colony System algorithm, pheromone is added only to arcs belonging to the global-best solution [9]. Advantages of such systems are inherent parallelism and can exhibit high level of robustness, scalability, fault tolerance and effectiveness on unquantified data along with simplicity of individual agent ([10],[11]).

II. Application of AI Techniques in Real-Time Systems

AI techniques are required to work continuously, deal with uncertain data, and handle both synchronous and asynchronous events in a predictable fashion with guaranteed response time when applied to real-time systems. It becomes critical when it is applied to operating system of the real-time system. For hard real-time systems it may not be advisable to use AI technique but definitely it can be used for soft-real time systems.

Fuzzy logic has been applied for scheduling of soft real-time systems [12]. It has been observed that deadline miss ratio has been decreased and overall throughput of the system has been increased.

Evolutionary approaches are not usually considered for real-time scheduling problems due to long computation times and uncertainty about the length of computation time. But because of inherent parallelism of genetic algorithm, it has been applied for scheduling of aircraft landing at Sydney airport on business day of the year [13]. The results show that high quality solutions can be computed in the time window between aircraft landings.

Soft real-time scheduling for general purpose client server systems also has been applied [14]. It has been taken care for the servers aware of their clients' quality of service without major restructuring. The ideas have been implemented in a system called Linux-SRT.

Soft real-time scheduling system with the aim to provide performance guarantee has been developed [15]. It is observed that predictive control scheduling can regulate the utilization and the deadline miss of real-time scheduling systems and show robustness when the actual execution time of tasks deviate from design value at run time.

ACO based scheduling algorithm for real-time operating systems has been proposed to achieve overall higher system performance and better throughput [16]. It has been discussed in more detail in the next section.

III. Application of ACO for Real-Time Systems

A reasonable way to measure the performance of a scheduling algorithm during an overload is by the amount of work the scheduler can feasibly schedule according to the algorithm. The larger this amount the better the algorithm. In real-time systems, deadline meeting is most important and we are interested in finding whether the task is meeting the deadline. Therefore the most appropriate performance metric is the Success Ratio and defined as [17]. During simulation, results are obtained with periodic tasks, measured in terms of Success Ratio and compared with Earliest Deadline First (EDF) algorithm in the same environment. It has been observed that the ACO based algorithm is equally optimal during underloaded conditions and it performs better during overloaded conditions. The results are shown in Fig. 1.

ACO based algorithm is good in all the conditions but it takes more time for execution as shown in Fig. 2. Scheduling time increase considerably in case of application of ACO based algorithm.

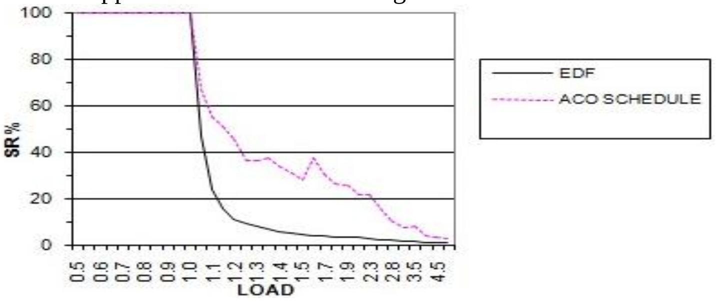

Fig.1. Comparison of Success Ration (SR %)

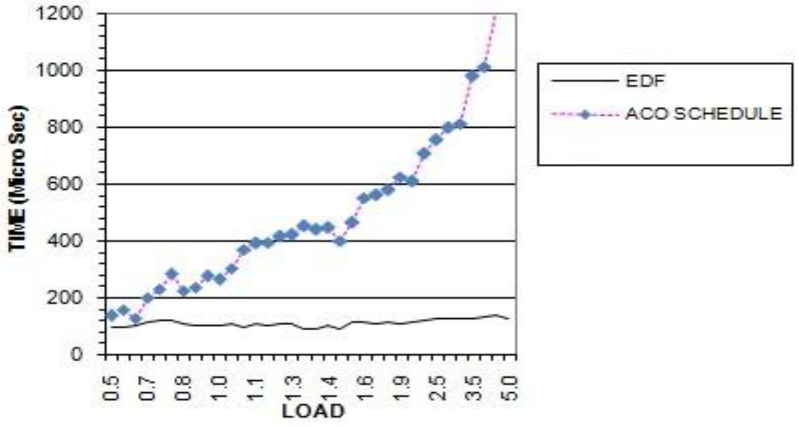

Fig.2 Comparison of Time taken for execution

IV. CONCLUSIONS

AI techniques cannot produce predictable results with guarantee. Real-time systems require predictable results that are why AI techniques were not suggested for them. But for soft real-time systems when the timing constraints are not stringent, AI techniques have been applied successfully. Different approaches with AI can improve the performance of the systems considerably and recommended for soft real-time systems.

REFERENCES

[1] K. Ramamritham and J. A. Stankovik, "Scheduling Algorithms and Operating Support for Real-Time Systems", *Proc. of the IEEE*, vol 82(1), pp. 55-67, 1994.

[2] C.L.Liu and L. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment", *Journal of ACM*, 20(10), pp. 46-61, January 1973.

[3] G.Saini, "Application of Fuzzy logic to Real-time scheduling", *Real-Time Conference*, 14th IEEE-NPSS. 2005

[4] C. D. Locke, "Best Effort Decision Making for Real-Time Scheduling", *Ph.d.thesis*, Computer Science Department, Carnegie-Mellon University, 1986.

[5] M.Dertouzos and K.Ogata, "Control robotics: The procedural control of physical process", *Proc. IFIP Congress*, 1974.

[6] A. Mok, "Fundamental Design Problems of Distributed Systems for the Hard-Real-Time Environment", *Ph.d.thesis*, MIT, Cambridge, Massachusetts, May 1983.

[7] E. Bonabeau and M. Dorigo and G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems", *Oxford University Press*, 1999.

[8] M.Dorigo and T. Stutzle, "Ant Colony Optimization", *The MIT Press*, Cambridge, MA, 2004.

[9] M.Dorigo and L.M.Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem", *IEEE Transaction on Evolutionary Computation*, vol. 1,April 1997, pp. 53–66.

[10] M. Dorigo and G. Caro, "The Ant Colony Optimization Metaheuristic in D.Corne, M. Dorigo and F.Glover(eds)", *New Ideas in Optimization*, McGraw Hill, 1999.

[11] V.Ramos, F.Muge, and P.Pina, "Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies", *In Second International Conference on Hybrid Intelligent System*, IOS Press, Santiago, 2002.

[12] G. Saini, "Application of fuzzy logic to real-time scheduling", 14th IEEE-NPSS, pp. 113-116, 2005.

[13] V. Ciesielski and P. Scerri, "Real-time genetic scheduling of aircraft landing times", IEEE World Congress on Computational Intelligence, pp. 360-366, 1998.

[14] D. Ingram, "Soft real-time scheduling for general purpose client-server systems", 7th Workshop on Hot Topics in Operating Systems (HotOS-VII), March 1999.

[15] J. Zhang and Y. Zou, "Predictive control for performance guarantee in soft real-time scheduling systems", Proceedings of the 6th World Congress on Intelligent Control and Automation, pp. 6944-6948, 2006.

[16] K. Kotecha and A. Shah, "Adaptive Scheduling Algorithm for Real-Time Operating System". *In Proceedings of 2008 IEEE World Congress on Evolutionary Computation (CEC 2008)*, pp. 2109-2112, June 2008.

[17] K. Ramamrithm, J.A. Stankovic and P.F. Shiah, "Efficient scheduling algorithms for real-time multiprocessor systems", IEEE Transaction on Parallel and Distributed Systems, vol. 1(2), pp. 184-194, 1990.