Global Journal of Engineering Design & Technology

Extended Abstract

Deadlock-Detection via Reinforcement Learning

Chen M

Abstract

Optimization of makespan in scheduling is a highly desirable research topic, deadlock detection and prevention is one of the fundamental issues. Supported by what learned from this class, a reinforcement learning approach is developed to unravel this optimization difficulty. By evaluating this RL model on forty classical non-buffer benchmarks and compare with other alternative algorithms, we presented a near-optimal result.

Keywords

Reinforcement learning; Optimization makespan; DL detection

Introduction

Due to buffer-less setting, deadlock (DL) occurs frequently in resource sharing environment and concurrent computing systems. A deadlock is a state in which each member of a group of actions is waiting for some other member to release a lock. [1] Once this DL state occurred, workflow would stack in a fixed loop and never discharged. Figure 1 present a typical scheduling problem: 3 jobs need to be operated on 3 different machines following different sequences, each machine can only operate one job each time. How to schedule jobs in specific sequence to minimize total makespan aka processing time without deadlock is a typical optimization problem. Due to limitation of resources, deadlock happened frequently, other than a feasible solution, to find the global optimal deadlock free solution is difficult. There are certain methods to solving deadlock problems: 1. Do nothing, 2. Kill the workflow, 3. Preempt and rollback. Other than kill the workflow, deadlock detection algorithms are more efficient in most cases and additionally, deadlock free scheduler would enable the realtime control for engineering system. Preventing or avoiding deadlock helping maintain system performance aka makespan stay at positive level. A simple head-tail scheduling example is present in Figure 2. The rectangle stands for resource Ri, symbol Pi stands for jobs, if rectangle is empty them means no job is operating on that resource.

Evaluation

There are 40 classical scheduling benchmark problems for testing. The design of these problems adopts complex structure to increase difficulty.

Additionally, if these systems are buffer-less, find scheduling will harder. Gantt chart can be drawn based on a DL-free timesheet obtained by each scheduling benchmark [27-31]. As shown in Figures 8 and 9, they present benchmark LA08 (15 × 5) and benchmark LA16 (10 × 10). We test the performance of our algorithm in this 40 benchmarks with backtracking counting's, we also compare the running time between with and without DL detection. Algorithms are written in Matlab.

Conclusion

Based on the ranking matrix, graph model and reinforcement learning, a new corresponding DL detection algorithm is proposed by us, and using that the author analyzed the general pattern of high-level DL detection problem based on discrete system, using the classicalforty benchmark problems. However due to the heavy computation, some work might took very long term, but this can be solved in time while the computation speed is exponential increasing. This algorithm is developed under the buffer less environmental which is much more difficulty compare to real world. Therefore, it is worth believing that our algorithm should be extended to other resource sharing systems. Based on this DL detection algorithms, relax some certain constrains new limited buffer DL detection algorithms can be developed and can be widely applied in the mechanical system, parallel computing system, and the future is quite bright

References

1. Coulouris GF, Dollimore J, Kindberg T (2005) Distributed systems: concepts and design. Pearson education, Chicago.

2. Minnameier C (2007) Local and global deadlock-detection in component-based systems are NP-hard. Information Processing Letters, Chicago 103: 105-111.

Mengmeng Chen
University of Central Florida, Orlando, USA, CHENMM@Knights.ucf.edu

3. Wysk RA, Yang NS, Joshi S (1991) Detection of deadlocks in flexible manufacturing cells. IEEE Transactions on Robotics and Automation 7: 853-859.

4. Dijkstra EW (2006) Een algorithme ter voorkoming van de dodelijke omarming. Chicago.

5. Chu F, Xie X (1997) Deadlock analysis of Petri nets using siphons and mathematical programming. IEEE Trans Robot Automat 13: 793-804.

6. Ezpeleta J, Colom JM, Martinez J (1995) A Petri net based deadlock prevention policy for flexible manufacturing system. IEEE Trans Robotics Automat 11: 173-184.

7. Fanti MP, Maione B, Mascolo S, Turchiano B (1997b) Low-Cost deadlock avoidance policies for flexible production systems. Int J Model Simulation 17: 310-316.

8. Fanti MP, Zhou MC (2004) Deadlock control methods in automated manufacturing systems. IEEE Trans on Systems, Man and Cybernetics - Part A: Systems and Humans 34: 5-22.

9. Gebraeel NZ, Lawley MA (2001) Deadlock detection, prevention and avoidance for automated tool shared systems. IEEE Trans on Robotics and Automation 17: 342-356.

10. Gen M, Cheng R (2000) Genetic Algorithms and Engineering Optimization. John Wiley & Sons.

11. Hsieh F, Chang S (1994) Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems. IEEE Trans on Robotics and Automation 10: 196-209.

12. Wu NQ (1999) Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model. IEEE Trans Syst Man Cyber Part C 29: 192-204.

13. Lawley MA, Reveliotis SA, Ferreira P (1998) The application and evaluation of Banker's algorithm for deadlock freebuffer space allocation in flexible manufacturing systems. International Journal of Flexible Manufacturing Systems 10: 73-100.

14. Taubin A, Kondratyev A, Kinshinevsky M (1998) Deadlock prevention using Petri nets and their unfoldings. Int J Manufact Technol 14: 750-759.

15. Reveliotis SA, Ferreira PM (1996) Deadlock avoidance policies for automated manufacturing cells. IEEE Transactions on Robotics and Automation 12: 845-857.

16. Viswanadham N, Narahari Y, Johnson TL (1990) Deadlock prevention and deadlock avoidance in flexible manufacturing system using Petri net models. IEEE Trans on Robotics and Automation 6: 713-723.

17. Yalcin A, Boucher TO (2000) Deadlock avoidance in flexible manufacturing systems using finite automata. IEEE Trans on Robot Auto 16: 424-429.

18. Xu G, Wu ZM (2004) Deadlock free scheduling strategy for automated production cell. IEEE Trans Systems Man and Cybernetics part A 34: 113-122.

19. Ramirez SA, Benhabib B (2000) Supervisory control of multiworkcell manufacturing systems with shared resources. IEEE Trans Syst Man Cybern part B 30: 668-683.

20. Reingold EM, Nievergelt J, Deo N (1997) Combinatorial

Mengmeng Chen
University of Central Florida, Orlando, USA, CHENMM@Knights.ucf.edu