International Journal of u- and e- Service, Science and Technology
Vol.9, No. 2 (2016), pp.353-362
http://dx.doi.org/10.14257/ijunesst.2016.9.2.34

Deadlock Avoidance Based on Graph Theory

Biyuan Yao, Jianhua Yin* and Wei Wu

College of Information Science and Technology, Hainan University
Haikou 570228, China
yaobiyuanyy@163.com, yinjh@hainu.edu.cn, wuweiisle@163.com

Abstract

Deadlock Avoidance remains to be a significant aspect of deadlock research. Modeling approaches based on Graph Theory provide effective strategy to solve this problem. We built avoidance model of directed graph and adjacent matrix via resource allocation graph, proposed three improved algorithm for deadlock avoidance and discussed the implementation of the program. The topology analyze of matrix storage information indicates entire optimization utilization results though certain key vertices and edges, this model verified by the banker's algorithm and Petri net finally.

Keywords: *graph theory, deadlock avoidance, directed graph, resource optimization, Petri net*

1. Introduction

Deadlock is a phenomenon due to the blocking of system resources among competing processes or communication. The problem is first proposed by Dijkstra in banker's algorithm research in 1965, when it was one of the tough issues facing computer operating systems, specifically, concurrent program design process. Deadlock describes a state of a collection of some parts trapped into an endless loop waiting for service which occupied by other parts, vice versa [1,2]. The main reason for deadlock was resource responses to request, allocation and release are random and time-varying. The key point to cope with the problem is to control the state of the system running and secure the state space from resource cycling-waiting, enable the entire system state activities acceptable and secured[3,4,5].

Deadlock can be expressed via directed system resource allocation graph accurately, including the nodes (represents the resource) and vertexes (represents all the possible changes of the system), thus enable to determine deadlock by resource allocation graph: it will deadlock with one instance of each resource graph; it may deadlock with multiple instances of each resource graph, it will not deadlock without loops [6,7].

Researches focused on deadlock testing, avoidance and release varies from different aspects did a good help for current scholars, while deadlock avoidance from graph theory is not common. Graph theory is well applied to establish a hierarchical model database, network structure model and the relational structure established by relational theory [8]. Therefore, this paper proposed feasible approach to deadlock avoidance based on graph theory.

The paper is mainly focused on the deadlock avoidance including graph theory, Petri net and directed graph, organized as follows: Several preliminary results and theories were introduced in Section I, in the second Section, directed/undirected graph is adopted to determine whether a system trapped into deadlock. Section III is the core simulation on deadlock avoidance consists of three approaches closely interconnected, specifically, involves the banker's algorithm model, spanning tree algorithm of directed/undirected graph and Petri net. The conclusion and related works are shown in the finally section.

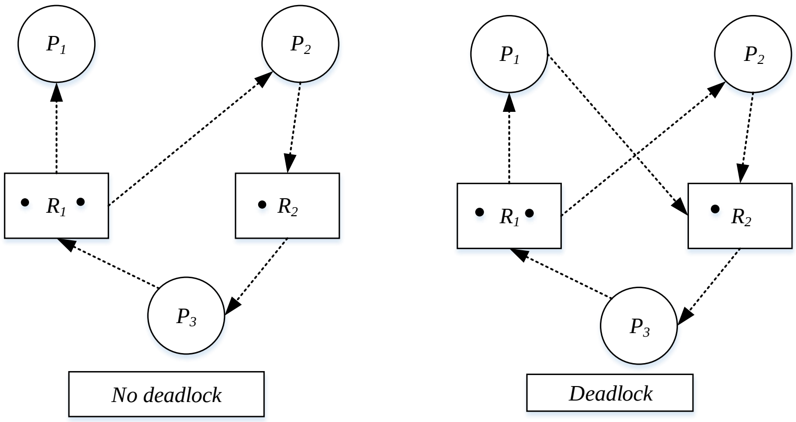
* Corresponding Author

ISSN: 2005-4246 IJUNESST

2. Deadlock Model Based on Graph Theory

2.1. Resource Allocation Graph

Resource allocation graph and waiting graph are two mainly methods to describe deadlock [9]. Resource allocation graph is a variant of waiting diagram with two kinds of nodes: process node and resource node. Resource allocation graph $G=(V,E)$, where the set of vertices $V=P \cup R$. $P=\{p0, p1,\ldots,pm\}$ is the set of processes, $R=\{r1, r2,\ldots,rn\}$ is a resource set, E is set of two kinds of edges: one is requesting edge (pi, rj), which indicates the process Pi requesting an unit of resource rj; the distribution edge (rj, pi) means the process pi occupies a unit of resource rj. Each edge is an ordered pair (pi, rj) or (rj, pi), where P represents the process, R represents a resource type. The resource allocation graph is a more powerful tool because waiting graph is limited for only one unit of resource in one kind of resource type. Moreover, in the resource allocation graph, the system may not be in a deadlock if there is directed loop, while it may mistakes in the waiting graph [10].



(a) (b)

Figure 1. Deadlock Demos of Resource Allocation Graph

Figure 1 indicates two resource allocation graphs. From (a), process P_1 obtains resource from R_1, and R_1 can be released to the process P_2; when process P_3 requests resource R_1, and process P_2 requests resource R_2 with the response resource of R_1, the process P_1 will release resource R_1 to the process P_3, by breaking the circuit, the process P_3 will release resource R_2 to process P_2, meanwhile, the process P_2 will release resource R_1 to process P_1, thus enable this system is not in deadlock status.

Process P_1 requests resource R_2 and been answered by R_1, process P_2 requests resource R_2 and been answered by R_1, process P_3 requests resource R_1 and been answered by R_2, process P_1 and P_2 requests resource R_2 and been answered by R_1, therefore, this system is a deadlock.

2.2. Deadlock Criterion Based on Directed Graph

Directed Graph is a geometrical diagram consists of directional edges and corresponding two vertices to the edge, such ordered pair could be expressed as $<v_i,v_j>$, and $<v_i,v_j>$ are two different directional edges. As is shown in Figure 2(a), they are two directed graph demos in terms of deadlock. To abstract the directed graph into simple expressions as

$P_1(R_2)\rightarrow P_2(R_1)\rightarrow P_1(R_4)$, $P_3(R_1)\rightarrow P_4(R_2)\rightarrow P_3(R_2)$, which is clearly shown in Figure 2(b),(c).

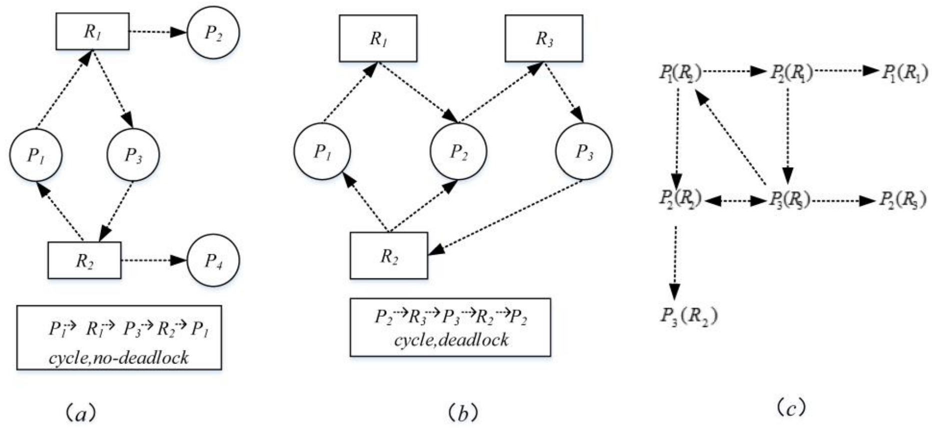


Figure 2. Directed Graph Demo with Loop

In Figure 2, Let G_1 is directed diagram, V_1 is a set of vertices which represents a particular database operation of a process, E_1 is the arc set which represents waiting process between two operations, thus meet:

$$\begin{cases}\{P_1(R_2),P_2(R_1),P_1(R_4),P_3(R_1),P_4(R_2),P_3(R_2)\}\in V_1\\ \{P_1(R_2)\to P_2(R_1),P_2(R_1)\to P_1(R_4),P_3(R_1)\to P_4(R_2),P_4(R_2)\to P_3(R_2)\}\in E_1\end{cases} \quad (1)$$

Any two vertices from V_1 are connected by only one arc, the directed graph G_1 related database operation experiences no deadlock. Similarly, in Figure 2(b),(c), Let G_2 is directed diagram, V_2 is a set of vertices which represents operation process, E_2 is the arc, the process running operation could be expressed as:

$$\begin{cases}\{P_1(R_2),P_2(R_1),P_1(R_1),P_2(R_2),P_2(R_3),P_3(R_3),P_3(R_2)\}\in V_2\\ \{P_1(R_2)\to P_2(R_1),P_2(R_1)\to P_1(R_1),P_2(R_1)\to P_2(R_2),P_2(R_2)\to P_3(R_3),P_3(R_3)\to P_2(R_2)\\ P_3(R_3)\to P_2(R_3),\ P_2(R_2)\to P_3(R_2),P_3(R_3)\to P_1(R_2),P_1(R_2)\to P_3(R_2)\}\in E_2\end{cases}$$
(2)

Therefore, $P_2(R_2)\to P_3(R_3)$,$P_3(R_3)\to P_2(R_2)$ indicates that there are two arcs between the edges, which enables G_2 is a undirected graph, and G_2 is expected to trap into deadlock.

2.3. Deadlock Avoidance Based on Directed Graph

Deadlock avoidance strategy algorithm based on directed graph is an effective way to identify deadlock, i.e. the controller use forward simulation in the process running to determine whether a deadlock occurs under some specified operations.

The resource-based deadlock is caused due to competition for resources [11,12]. There are four necessary conditions result in resource-based deadlock: i) Mutually exclusive conditions, the resource is occupied certain tasks exclusively at a time, if other tasks that request the resource is bound to be rejected until certain tasks complete operation and release the resource. ii) Request and maintain conditions, certain tasks request more resource and the tasks have occupied part of resource, meanwhile, the test resource has been allocated to other tasks, thus lead to deadlock because of request denied and the resource occupied unreleased. iii) Non-deprived conditions, similar to ii), those resource has been allocated could not be released until the process related finished. iv) Dead loop, there is a process queue which waiting for the running process forms a loop.

The basic idea is to relieve the deadlock is to eliminate loops from undirected graph to a directed graph. As long as specific deadlock or loop has been detected, the algorithm will cut out certain process, withdraw application command from the system to release resource for further utilization. As a result, other processes are allowed to send request of resource, which makes the system stable and effective.

As is discussed above, two ways adopted to avoid deadlock including resource application control and Petri structure control. For the first one, we try to simulate the result of permit the request from processes, if deadlock will be leaded, then the request is bound to be rejected. Alter the structure of Petri net to adapt request is an adaptive control model based on adding more signal monitors and controllers.

3. Modeling and Simulation

3.1. Banker's Algorithm

Banker's algorithm is based on bank lending system allocation strategy to determine and ensure the safe operation of the system. From the directed graph view, we decompose the directed graph model into more sub figures, thus enable us to analyze the security of each one by ordered distributed algorithm, it checks each request to check if it will lead to an unsafe condition when they added to the system [13,14]. However, this method requires total resource amount before process running. The flow chart is shown in Figure 3. For instance, there are three kinds of resources (*A*,*B*,*C*) and five processes (P_1, P_2,…,P_n) in t_0, the detailed situation is shown in Table 1.

Table 1. System with 3 Types of Resources and 5 Processes

	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P_0	7	5	3	0	1	0	7	4	3	3	3	2
P_1	3	2	2	2	0	0	1	2	2			
P_2	9	0	2	3	0	2	6	0	0			
P_3	2	2	2	2	1	1	0	1	1			
P_4	4	3	3	0	0	2	4	3	1			

For the system described above, our prior concern is if it is stable. P_1 will be allowed in consideration about the accessible resources of P_1 is above its request, to recollect sources occupied by P_1, the total accessible resource of the system would up to (5,3,2) from the sum of (3,3,2) and (2,0,0). Similarly, as to P_3, after series of processing and resource release, the total accessible resource of the system would up to (7,4,3) from the sum of (5,3,2) and (2,1,1). Then to P_0, it will be (7,5,3) from the sum of (7,4,3) and (0,1,0), P_2, it will be (10,5,5) from the sum of (7,5,3) and (3,0,2); finally, P_4 could be added to the system for resource allocation, when all processes are finished, the total reminder system resource is (10,5,7). Thus produces a secured sequence: $P_1 \rightarrow P_3 \rightarrow P_0 \rightarrow P_2 \rightarrow P_4$.

3.2. Graph-theoretical Algorithm

Graph-theoretical algorithm plays an important role in computer science, it provides for a lot of problems are a simple and effective systems modeling method. Graph-theoretical algorithm steps:

Step 1: Judgment graph is a directed graph or undirected graph.

Step 2: According to the request and allocation of resources and directed graph and undirected graph adjacency matrix, remember as *A*, *B*. By adjacency matrix symmetry is easy to diagnose whether the deadlock, and then to a deadlock.

Step 3: By adjacency matrix to survive as tree adjacency matrix, correlation matrix, respectively for the A_1, A_2, B_1, B_2. Adjacency matrix and incidence matrix are representative of this graph topology. The tree's adjacency matrix is used to store the relationship between the nodes in the graph.

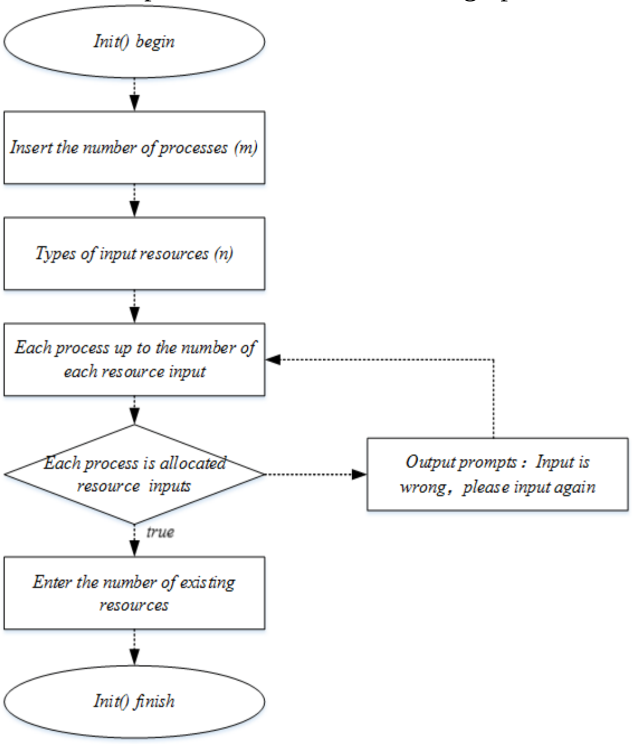


Figure 3. Flow Chart of Deadlock Analyze based on Banker's Algorithm

Information to a graph is stored by an adjacency matrix. For an undirected graph, adjacency matrix is symmetrical along the main diagonal, while it may not symmetrical along the main diagonal for a directed one.

For an undirected graph adjacency matrix, if edge [*i*][*j*]=1, then there is an edge between vertex *i* and *j*, i.e., Elements value in row/column *i* equal to 1 will be counted as degree of vertex *i*. For directed graph adjacency matrix, if edge [*i*][*j*]=1, then there is an directed edge from vertex *i* to *j*, i.e., Elements value in row/column *i* equal to 1 will be counted as degree of vertex *i*.

Algorithms to directed and undirected graph are likely the same. We choose a (directed) edge optionally, record vertex set *S* to it, remove all directed edges between the vertices in this set to generate matrix S_1, search for directed edges associated with vertex from S_1, and repeat the procedure before until S_i contains all vertices, thus the edges constitute a spanning tree.

As is illustrated in Figure 1, let *A*, *B* are adjacency matrixes A_1, B_1 signifies the adjacency matrixes of spanning trees. The adjacency matrix *A*, *B* could be expressed as:

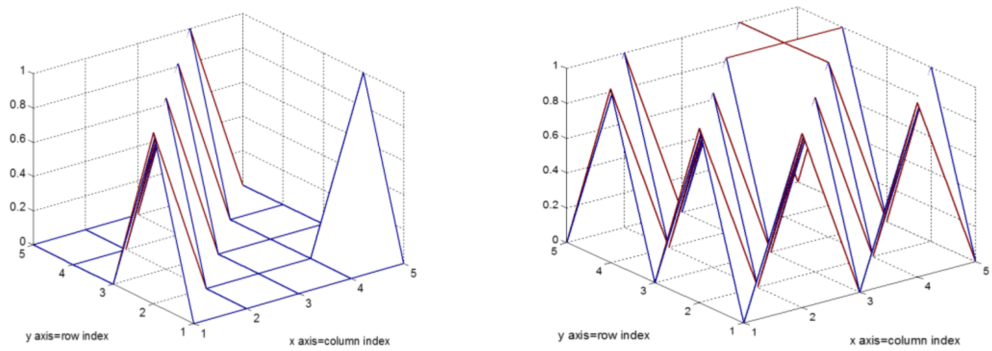
$$A=\begin{bmatrix}0&0&0&0&0\\1&0&0&0&1\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{bmatrix},B=\begin{bmatrix}0&1&0&1&0\\1&0&1&0&1\\0&1&0&1&0\\1&0&1&1&1\\0&1&0&1&0\end{bmatrix} \quad (3)$$

And the derived adjacency matrixes of spanning tree is:

$$A_1=\begin{bmatrix}0&0&0&0&0\\1&0&0&0&1\\0&0&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{bmatrix},B_1=\begin{bmatrix}0&0&0&1&0\\1&0&1&0&1\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\tag{4}$$

Figure 1(a) is directed and without weights, let the elements be 0 or 1 here, the output degree of vertices $i+1$ is equal to the number of elements in row i which equal to 1, similarly, the input degree of vertices $i+1$ is equal to the number of elements in row i which equal to 14, the degree of vertexes are:

$$A:\begin{cases}i=3, & d_{out}=2;\\ i=4,5 & d_{out}=1\,,\\ i=2,3,4,5 & d_{in}=1.\end{cases}\qquad B:\begin{cases}i=1,3,5, & d=2\\ i=2, & d=3;\\ i=4, & d=4.\end{cases}\tag{5}$$



(a) (b)

Figure 4. Correlation Analysis Diagram of Directed/undirected Graph

Figure 4(a) is directed and will not experience deadlock, figure 4(b) is undirected and will experience deadlock, the results consistent with the discussion in part *A*, section II.

3.3. Petri Net

Petri net is a graphics-based model through the formal language of mathematics, it studies structural and behavioral of actual systems to describe all changes that may occur [14]. Correlation matrix is one of the main analysis of Petri nets, it is mainly used in the form of a matrix to represent the relationship between each alternative and the importance of the evaluation indicators and programs on specific indicators of valuation.

The correlation matrix of *A* could be achieved as:

$$A_2=\begin{bmatrix}0&0&0&0&0\\1&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\-1&0&0&0&0\end{bmatrix},\quad B_2=\begin{bmatrix}1&1&0&0&0&0&0\\1&0&1&1&0&0&0\\0&0&1&0&1&0&0\\0&1&0&0&1&1&1\\0&0&0&1&0&0&1\end{bmatrix}\tag{6}$$

From A_2, each row reflects the connection between the vertexes of each edge, let it be defined as a vertex vector V, each column reflects the connection between the edges of each vertex, let it be defined as the edge vector E. From vertex 2, $V_2 = (1,0,0,0,0)$, wherein the first value 1 represents vertex 2 is the start point of edge 1, and the rest value 0 means there is no connection between vertex 2 and other vertexes. Thus, the topology of vertices and edges of the resource allocation graph is straightforward.

Correlation matrix B_2 has 4 rows and 7 columns, the row vector is apparently linearly independent, i.e., B_2 is full rank matrix. Thus, the resource allocation matrix could associated with correlation matrix. The correlation matrix of directed/undirected graph can be expressed in checkerboard in Figure 5:

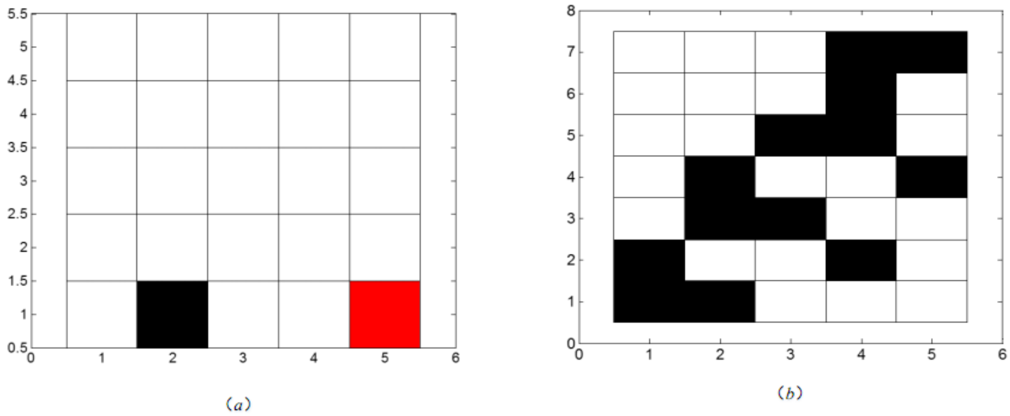


Figure 5. Correlation Matrix of Directed/Undirected Graph Expressed in Checkerboard

To do this, we use the following examples of Petri nets can be used to calculate Matlab quickly find and control network nodes will be described in a deadlock. Most elements from Petri net model equal zero, which could be stored in computer memory easily. A testing sample about auto sales machine is experimented below.

Let an auto sales machine provides bottled drink worth 1.5 or 2.0 *RMB*, and coins of 0.5 and 1.0 are accepted. Let P represent price nodes set, T is conversion operations set (pay coins, claim goods, refund, etc.), M is marking sets, the status flow is depicted in Fig. 6.

As in indicated in Fig. 7 (a) and (b), for a multiple-tasking sales machine monitoring station, if the resources application decreased dramatically at 40 second, the system could track this trend, and remains some margin above the application resources in case of emergency. But when the user application weakened, it hold its capacity not below the decreasing users, thus to prevent deadlock in advance, see point *A,B,C,D,E,F,G,H*. However, if the user capacity altered to an increasing way, the system may trapped into deadlock, but as system volume is higher than 4,000 machines, the system would remain stable in the whole period.

Through systemic modeling on auto sales machine monitoring station, the graph theory model receives worthy results even the user fluctuate dramatically. Thus enhanced the system stability and robust.

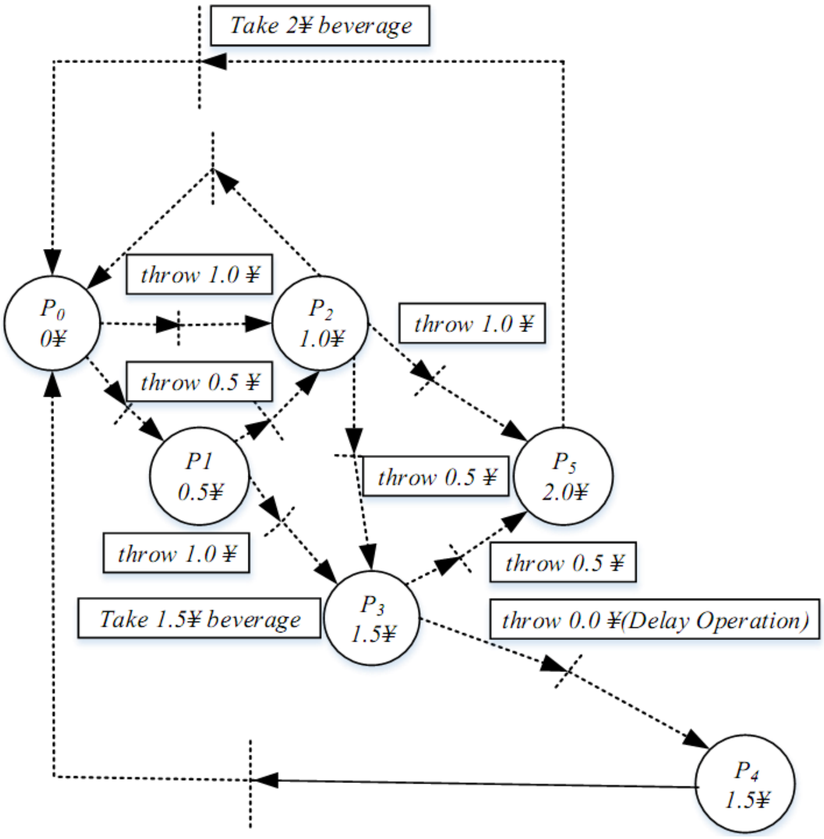


Figure 6. Petri Model of Auto Sales Machine

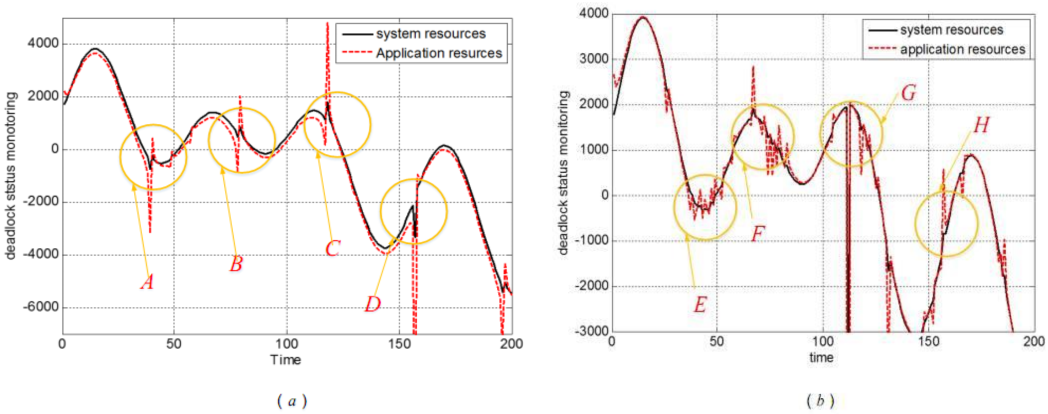


Figure 7. Deadlock Status Monitoring based on Graph Theory

4. Simulation Analysis and Conclusions

Directed graph model has been established and well applied for bankers' algorithm in part *A*, section III, this approach requires less computational time than traditional centralized online algorithms and guarantee the same flexibility. For a directed graph model, we decomposed it into several subsystems, enable it controllable with additional criteria and strategies. We also adopt matrix to storage resource allocation information based on the resource allocation graph theory, to analyze the topology and express the entire resource allocation map interface between the vertices and edges, effective control of key parameters to achieve the overall calculation optimization.

5. Related Works

Graph theory method is a simple and intuitive tools in terms of deadlock avoidance for describing the interaction between the vertices and edges, which makes the graph theory approach could be applied in complex deadlock related systems effectively, however, it cannot express manufacturing systems with multiple resource requests.

Directed graph is an effective and intuitive interactive approach for relationship modeling between processing tasks and resources, it identifies deadlock effectively and derived some strong adaptability deadlock avoidance strategies [15]. Petri nets as an important graphical modeling and digital analysis tool for distributed systems based on rigorous mathematical theory. For a network model, the use of Petri can control common deadlock enable the system in good operation, Disadvantages Petri net model is the model more complex which need to be calculated using software programming to save time and reduce costs in real circumstances.

Acknowledgements

This research work is supported by National Natural Science Foundation of China (Grant No. 11161016).

References

[1] S. M. Zhou, "Feedback Number of Honeycomb Networks", Chinese Journal of Engineering Mathematics, vol. 28, no. 2, **(2011)**, pp. 260-264.
[2] B. X. Zhu, "The least eigenvalue of a graph with a given domination number", Linear Algebra and its Applications, vol. 437, no. 11, **(2012)**, pp. 2713-2718.
[3] M. Nath and S. Paul, "On the distance spectral radius of trees", Linear and Multi-linear Algebra, vol. 61, no. 7, **(2013)**, pp. 847-855.
[4] M. Aouchiche and P. Hansen, "Two Laplacians for the distance matrix of a graph", Linear Algebra and its Applications, vol. 439, no. 1, **(2013)**, pp. 21-33.
[5] Y. Wang and B. Zhou, "On distance spectral radius of graphs", Linear Algebra and its Applications, vol. 438, no. 8, **(2013)**, pp. 3490-3503.
[6] D. Breuker, P. Delfmann and H. A. Dietrich, "Graph theory and model collection management: conceptual framework and runtime analysis of selected graph algorithms", Information Systems and e-Business Management, **(2014)**, pp. 1-38.
[7] H. Lei, K. Xing and L. Han, "Deadlock-free scheduling for flexible manufacturing systems using Petri nets and heuristic search", Computers and Industrial Engineering, vol. 72, （**2014**）, pp. 297-305.
[8] F. Wang, K..Y. Xing and X. P. Xu, "An enumeration method to find a deadlock structure", Remote Sensing and Smart City, vol. 64, （**2015**）, p. 173.
[9] J. Sifakis, S. Bensalem and S. Bliudze, "A Theory Agenda for Component-Based Design", Software, Services, and Systems. Springer International Publishing, （**2015**）, pp. 409-439.
[10] Y. F. Hou, Z. W. Li and M. Zhao, "Extraction of elementary siphons in a class of generalized Petri nets using graph theory", Engineering Computations, vol. 31, no. 2, （**2014**）, pp. 331-352.
[11] R. Akhtar, A. B. Evans and D. Pritikin, X"Representation numbers of complete multipartite graphs", Discrete Mathematics, vol. 6, no. 312, （**2012**）, pp. 1158-1165.
[12] G. Wang, Y. Wang and J. Ren, "Graph Algorithm Theory, Implementation and Applications", Peking University Press, **(2011)**.
[13] H. Wang, Q. Huang, C. Li and B. Zhe, "Graph Theory Algorithms Based on MATLAB", Beijing University of Aeronautics and Astronautics Press, **(2010)**.
[14] W. Stallings, "Operating Systems Internals and Design Principles", Electronic Industry Press, **(2010)**.
[15] X. Wang, Q. Wang and Y. Guo, "Deadlock Detection Based on the Parallel Graph Theory Algorithm", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering, **(2013)**; Zhejiang,China.

Authors

Biyuan Yao, She is a Master of Hainan University, her research interest covers Graph Theory with Applications.

Jianhua Yin, He is a Professor of Hainan University, MS supervisor. His research interest: Graph Theory with Applications.

Wei Wu, He received Master degree from Hainan University. His research interest covers data and knowledge engineering.