RESOURCES ALLOCATION CONTROL IN FLEXIBLE MANUFACTURING SYSTEMS USING THE DEADLOCK AVOIDANCE METHOD

Francisco Yastami Nakamoto, francisco.nakamoto@poli.usp.br, yastami@fsa.br
University of São Paulo, Escola Politécnica, São Paulo, SP, BRAZIL
Centro Universitário Fundação Santo André, Santo André, SP, BRAZIL

Paulo Eigi Miyagi, pemiyagi@usp.br
University of São Paulo, Escola Politécnica, São Paulo, SP, BRAZIL

Diolino José dos Santos Filho, diolino.santos@poli.usp.br
University of São Paulo, Escola Politécnica, São Paulo, SP, BRAZIL

Abstract. *The Flexible Manufacturing Systems (FMS) belong to class of productive systems in which the main characteristic is the simultaneous execution of several processes and sharing a finite set of resource. The process is defined as being a sequence of steps or activities. These activities are established a priori as being activities of transformation and/or transportation in which the aggregate value of the product or a service rendering if increases by each concluded activity. The FMS belongs to a class of Discrete Events Dynamic Systems (DEDS). The discrete state of these systems evolves when the events occurs, in other words, the occurrence of event causes an abrupt state transition in the system. Nowadays, the FMS must attend the demand of the market needs for personalized products. Consequently the product life cycles tends to be shorter and a greater variety of products must be produced in a simultaneous manner. Thus, the flexibility is an important factor of the FMS to execute with effective and efficient form their functions. The FMS efficiency is one of the more important factors studied by the researchers. The aspect of the processes optimization involves three questions: (i) the access to the new technological resource, (ii) the optimization of the resources allocation and (iii) the need to updating of the control system every time arises a demand of a new product or new service rendering. The main objective is improves the overall capacity of the plant. As a result, it is very important the control of such systems considering the optimization of the resources allocation. In previous work was presented the case study that involves one instance of each resource and the systematization of the resource control design. In this work will be presented a case study of multiple instances of resource in FMS, and an improvement of the algorithm to find the closed wait cycles with adjacency list representation, and present the new algorithm for generation of the control additional rules for avoid deadlock in FMS.*

Keywords: *Flexible manufacturing systems, Petri net, discrete event dynamic system, deadlock avoidance, system control.*

1. INTRODUCTION

The Flexible Manufacturing Systems (FMSs) must attend the demand of the market needs for personalized products. Consequently the product life cycles tends to be shorter and a greater variety of products must be produced in a simultaneous manner. The FMSs are integrated systems of computer numerically controlled machine tools, an automated material handling system, distributed buffer storage sites and a computer supervisory controller for monitoring the status of jobs, part routing and machine job selections (Stecke, 1986; Lewis *et al.*, 1998). The FMS is characterized by the simultaneous execution of several processes, sharing the same set of resources. The FMS belongs to the class Discrete Events Dynamic Systems (DEDS) (Ho and Cao, 1991; Cassandras, 1993), consequently, the properties such as indeterminism, conflict and parallelism are present in this kind of systems. Besides, the dynamic behavior of FMS is based on the occurrence of events, which causes an abrupt transition from one discrete state to another. In FMS, the control of a single process is relatively simple: It should guarantee the sequence of the activities (or stages). However, in a production system with simultaneous execution of several processes and sharing a finite set of resources, the systems may eventually “die”. That is called deadlock or deadlock state (Banaszak and Krogh, 1990; Viswanadham *et al.*, 1990; Cho, 1993, Kumaran *et al.*, 1994). This phenomenon occurs when the flow of processes is permanently interrupted and/or when the activities of the processes can not executed. The deadlock occurrence in FMS is due to the four conditions (Banaszak and Krogh, 1990; Cho, 1993; Fanti *et al.*, 1997; Santos Filho, 2000a): mutual exclusion, retention while waits, no preemption and circular wait condition. The deadlock will occur if all four conditions are true at same time. Thus, it is enough guarantees that at least one of the conditions above is never satisfied. In a FMS, the first three conditions can always occur, that is, two or more processes cannot use the same resource simultaneously and only the process that allocates a resource can release it. Consequently, the retention occurs while the process waits for the available resource. Therefore, circular wait condition remains as the condition that can be controlled.

There are three basic approaches to deal with the deadlock problems (Fanti *et al.*, 2000; Santos Filho 2002):

(i) Deadlock detection and recovery: detect deadlock occurrences and restore the systems operations with recovery procedures;

(ii) Deadlock prevention: prevents circular wait conditions using offline strategies;

(iii) Deadlock avoidance: prevents deadlock situation applying online policy control of resource allocation.

Several works propose solutions to board the deadlock problem through a policy control of resource allocation (Banaszak and Krogh, 1990; Viswanadham *et al.*, 1990; Cho, 1993, Kumaran *et al.*, 1994; Fanti *et al.*, 1997; Fanti *et al.*, 2000; Santos Filho, 2000a; Hsieh, 2000; 2002). The implementation complexity of these methods is proportional to the complexity of the system (Edmonds, 1995), and it may become unfeasible due to the great computational effort required. Another important aspect is related to the resources allocation control. Banaszak and Krogh (1990) considered a class of PN model and proposed the Deadlock Avoidance Algorithm (DAA) for FMSs with concurrently competing process flows. The DAA allows an event to take place when certain resource conditions are satisfied under the current systems state, in which the proposed control policy is sufficient for avoiding deadlock. Viswanadham *et al.* (1990) proposes the implementation of deadlock prevention by using the reachability graph on Petri net models and proposed the algorithm for real-time controller using the deadlock avoidance method in Petri net-based model. In Banaszak and Krogh (1990) and Viswanadham *et al.* (1990) the control of the processes and the control of the resources are handled at the same level. Fanti *et al.* (1997) proposes control strategies to be applied to the resources control using the graph-theoretic framework. They introduce two types of digraphs, named working procedure digraph and transition digraph. In another paper (Fanti *et al.*, 2000), the authors present the connections between the graph-theoretical approach and Petri net models to deal with the deadlock problems in FMS with deadlock avoidance method. Banaszak and Polak (2002) present a study of relationship between system resource capacities and an initial state with priority rules allocation in sequential cyclic processes. Basically, a control policy is established that restricts a process entering a buffer and the influence of initial state with priority rules allocation to the buffer capacity. Ezpeleta *et al.* (2002) propose an extension of the Banker's algorithm with a class of Petri nets model, called S*PR, able to deal with the deadlock problems in sequential resource allocation systems (S-RAS) with routing flexibility and the use of multiple instances of different resources per activity. Moreover, Ezpeleta and Recalde (2004) propose the deadlock avoidance approach in nonsequential resource Allocation systems (NS-RAS). This approach is based on partial reachability graph and Petri net models with an adaptation of the Banker's algorithm in flexible part routing. To solve the problem, the authors transform the control of a NS-RAS into the problem of controlling a S-RAS. In Hsieh (2000), the author proposes a framework for generating a deadlock avoidance control synthesis algorithm by using the controlled Petri net model, called controlled assembly Petri net (CAPN) model. The algorithm analyzes the liveness conditions, vulnerability and reconfigurability of CAPN for dealing with the addition and removal of processes, as well as identifying the possible fragile part in the manufacturing systems to provide robustness to them. Moreover, the same author (Hsieh, 2002) presents a model for conflict resolution and control of Holonic Manufacturing Systems (HMS) based on cooperative agents. The modern manufacturing system requires being adaptive to deal with dynamic situations such as changes of product specifications, machine breakdown, emergency orders and other kinds of disturbances. In this context, the holonic manufacturing systems were proposed to deal with dynamic changes in these systems. Hsieh (2002) presents a multi-agent framework to reorganize the available resource to achieve the production goal and an effective coordination mechanism to resolve agents conflict situations. The HMS consists of resource holons, product holons, and order holons, in which the interactions between agents are described by a graph theoretical model. This model is called collaborative commitment graph to represent the collaborative network among the agents. The author proposed the conversion of collaborative networks into Petri nets, called collaborative Petri nets (CPN). Santos Filho (2000a; 2000b) proposed the deadlock avoidance method for FMS based on the partition of the control system into two levels of control: (i) Control of processes and (ii) Control of resource. The author uses the resource allocation graph (RAG) (Cormen *et al.*, 2003) and the enhanced mark flow graph (E-MFG) (Santos Filho, 1995; 2000a, 2000b) to model the system control of FMS. The E-MFG is an extension of mark flow graph (Miyagi, 1996) and deduced from the Petri nets model (Peterson, 1981; Reisig, 1985; Murata, 1998). This structure enables the process control through a restriction model that determines how the resources should be allocated during the process evolution by the resource control rules. Adopting the same control structure of Santos Filho (2000a; 2000b), Nakamoto *et al.* (2002a) presented the algorithm for determination of circular wait condition in RAG, called circular wait loops (CWL). In Nakamoto et al (2002b) was presented the algorithm for determination of the resources allocation rules (additional rules of control) and in Nakamoto *et al.* (2003a; 2003b; 2003c) was introduced the systematization of the project of the production system control with one instance of each resource.

This work aims to introduce the utilization of another data structures to represents a RAG model and the respective algorithms for determination of CWL and the generation resources allocation rules for avoid deadlock in FMS with multiple instances of resources.

2. DESIGN THE CONTROL SYSTEMS OF FMS

The difficulty of modeling the system control of FMSs is the number of reachable state that can be exponential. Such problems depend on the number of processes, resources and other variable of the manufacturing systems. The complexity is the property of representation in which the global behavior is hard to be formulated even when all information about the components and their interrelations are provided (Edmonds, 1995; Palazzo and Castilho, 1998; Calinescu *et al.*, 2000). In other words, the complexity is a property that the models when certain behavior is imposed on FMS (Santos Filho, 2000a). Imposing a behavior for a single process is relatively simple, because it involves only the guarantee of the sequencing of the predetermined activities of the process. In a global context, it is not possible to determine the behavior of the system in a purely sequential manner. The complexity of the global process in FMS is due to two levels of indeterminism (Santos Filho, 2000a):

(i) Indeterminism regarding time: It is not possible to determine when a certain event is going to occur;
(ii) Indeterminism regarding the sequence of events: It is not possible to determine which event precedes another. That situation is due to the execution of simultaneous processes sharing a finite set of resources. Consequently, it is not possible to prescribe the system behavior as a whole since several processes are executed simultaneously.

The controller design adopted in this work is based on three fundamental aspects (Santos Filho, 2000a):

a) The restriction models, i.e., it is not possible to determine all reachable states of the system but it is possible to avoid the undesirable state;
b) The anthropocentric control systems (Ito, 1991), i.e., the human element participates in the requisite specification, development, implementation and monitoring and, whenever necessary, executes actions which interfere in the dynamics of the system;
c) The partition of the control system into two modules: Control of processes (CP), responsible for control of the sequencing of activities of the processes, and, Control of Resource (CR), responsible for managing the utilization of the resources by the processes.

Basically, the CR monitors the stages of the processes regarding the utilization of resources according to resource allocation policy (set of additional control rules). Both control modules have distinct semantics, i.e., the CP request to CR the utilization of a certain resource. If this resource can be allocated, the CR changes the resource status, thus, the resource can not be used by other processes, and, the CR sends an authorization to CP indicating that the resource is available to be allocated. The CR and CP is generated of the following form (Nakamoto *et al.* (2003a; 2003b; 2003c):

- ❖ Obtaining the individual RAG of each process (Fig. 1a);
- ❖ Obtaining the global RAG from the individual RAG (Fig. 1b);
- ❖ Determining the CWL;
- ❖ Generating the Control Rules;
- ❖ Generating the E-MFG from global RAG and (Fig. 1c);
- ❖ Inserting the Control Rules (Fig. 2);

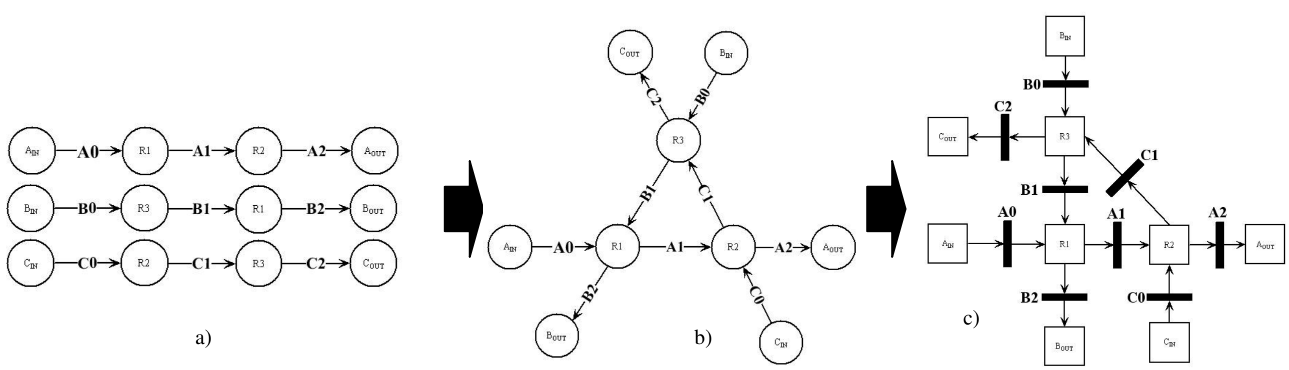


Figure 1 Example of individual and global RAG.

In figure 2 show the architecture of the control system and the communication of CR and CP with inhibitor arcs of E-MFG.

3. DATA STRUCTURES AND ALGORITHM

The computational resources are fundamental tools in several systems for accomplishing the data processing, data storage and the control execution. According Ziviani (2004), the development of the computers program is basically built with the data structure and the algorithms. The form in which the data will be processed depends on the representation of these data. The computational evolution allows the access for faster processors and a larger storage of data with reduced costs. These facts allow the complex data structure implementation and algorithms, which before were unfeasible. In this section will be introduced the data structures adopted for determine de CWL and of the additional rules of control.

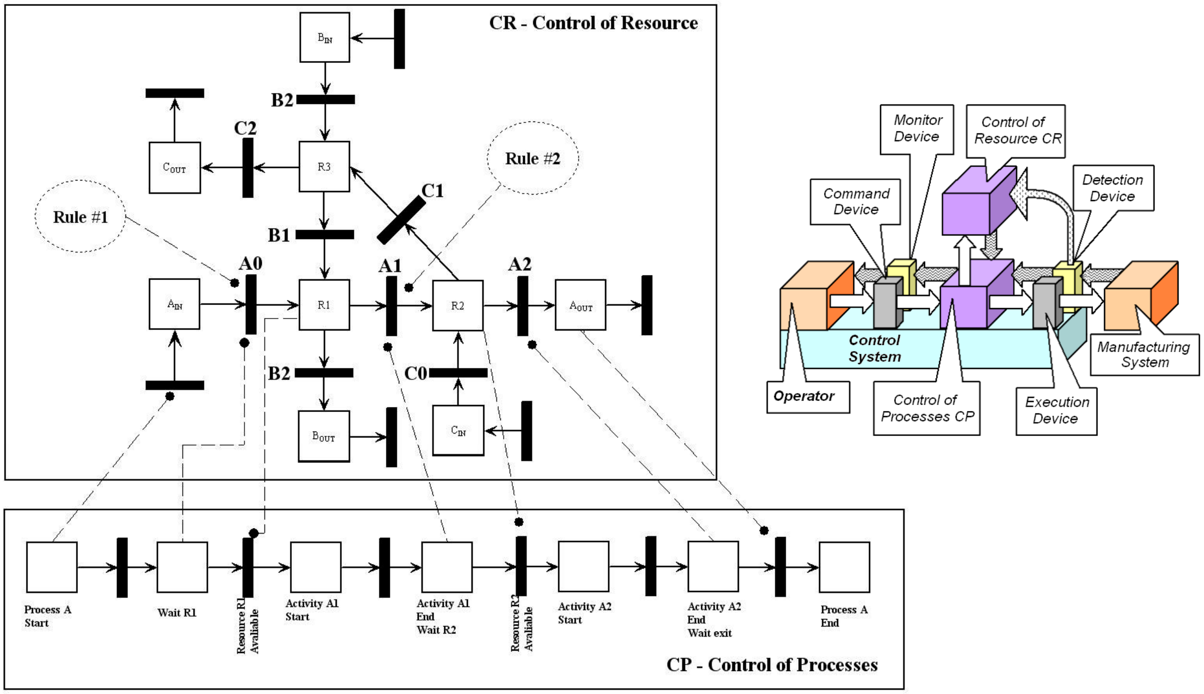


Figure 2 The architecture of the Control System.

3.1 Determination of CWL

The Resource Allocation Graph (RAG) is an unmarked directed graph (Cormen et al, 2003) which represents the resource allocation of the processes in FMS. The RAG **G = (R, A)** is a non empty set of resources (nodes) **R = {r_1, r_2, r_3, …, r_n}** and set of ordered pair of arcs **A**. An arc **a_{ij} = (r_i, r_j)**, were **r_i** and **r_j** are elements of **R**. The RAG is used to determine the condition of CWL. For example, in Figure 2a presents an example of RAG with three CWLs: {r_1, r_2, r_4 and r_3}, {r_1, r_4 and r_3} and {r_1, r_2, r_5, r_4 and r_3}.

In this work is presents the representation of a RAG **G = (R, A)** using the adjacency-list (Cormen et al, 2003) instead of the adjacency matrix proposed in Nakamoto *et al.* (2002a). The adjacency list consists of an array **Adj** of **|R|** lists, one for each resource in **R**. For each **r ∈ R**, the adjacency list **Adj[r]** contains all the resource adjacent to **r** in **G**. In Figure 3 presents an example of adjacency matrix representation and adjacency list representation.

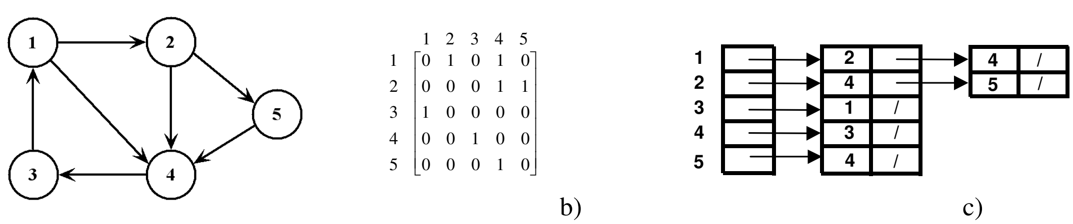


Figure 3 Example of RAG (a), the adjacency matrix (b) and adjacency list representation (c).

When the adjacency list is given by tabular form, this representation is denominated Star (Drozdek, 2002), in which it can be direct (ForwardStar) or inverse (ReverseStar). The Figure 4 introduces the direct and inverse representation of the example RAG in Figure 3.

The great advantage of the utilization of the adjacency list representation is regarding the complexity of the algorithm, in comparison to the adjacency matrix adopted previously. While the adjacency matrix has the complexity of **$\Theta(R^2)$**, the adjacency list has the complexity of **$\Theta(R+A)$** (Cormen et. al, 2003; Drozdek, 2002; Ziviani, 2004), consequently, the adjacency list representation allows a better performance of computational processing time to determine the CWLs in RAG.

The algorithm executes a depth-first search (Cormen et. al, 2003) in an adjacency list of RAG (searching CWLs). Basically, the strategy is to search the most depth in the RAG. The resources (nodes) are explored out of the most recently discovered which still has unexplored arcs leaving this resource. When all the adjacent arcs are explored, the algorithm returns to the predecessor node and explores the rest of adjacent resource not yet explored. A CWL is found if the discovered resource was visited previously. The Figure 5 presents the proposed algorithm and the auxiliary functions.

i	Origin	Destiny	Arc
0	1	2	a_{12}
1	1	4	a_{14}
2	2	5	a_{25}
3	2	4	a_{24}
4	3	1	a_{31}
5	4	3	a_{43}
6	5	4	a_{54}

Origin	ForwardStar
0	-1
1	0
2	2
3	4
4	5
5	6
6	7

i	Origem	Destiny	Arc
0	3	1	a_{31}
1	1	2	a_{12}
2	4	3	a_{43}
3	1	4	a_{14}
4	2	4	a_{24}
5	5	4	a_{54}
6	2	5	a_{25}

Destiny	ReverseStar
0	-1
1	0
2	1
3	2
4	3
5	6
6	7

Figure 4 Examples of ForwardStar and ReverseStar.

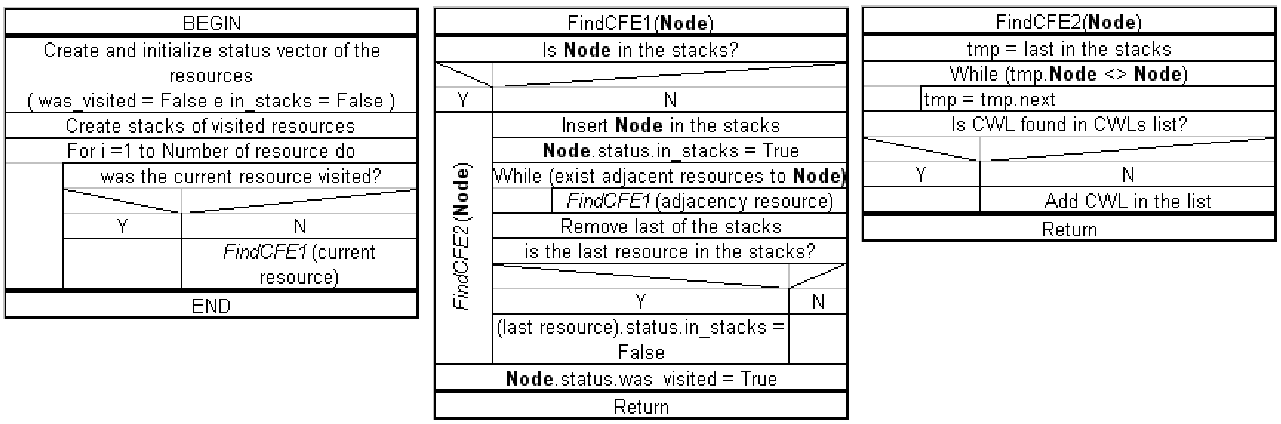


Figure 5 Algorithm for CWLs determination in an adjacency list of RAG.

A CWL is discovered when the algorithm finds a resource visited previously. Then it verifies if CWL was discovered previously before inserting in the list of CWL. The algorithm uses the auxiliary data structures to find CWL and insert in the list: data structure stack and status vector of resource.

3.2 Determination of Control Rules of Resource Allocation

The following notation will be used to determine the control rules of resource allocation:

- P_C : Number of resources used by the processes which belong to the c-th CWL;
- Q_C : Total capacity of available resources of the c-th CWL.

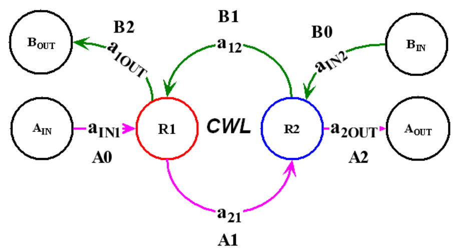
The control rules are deduced from the list of CWLs. Based on flow-in-suppression deadlock avoidance policies (focused pre-deadlock condition), for each CWL, the algorithm calculates the capacity of available resource. The pre-deadlock condition is characterized when just one resources is available and the others resources are occupied by processes that will stay in CWL. The rules inhibit the entrance of the processes in CWL, i.e., the last available resource can not be used for avoid the deadlock situation (Fig. 6a). The algorithm creates "IF... THEN..." type production rules. The condition "IF" describes the pre-deadlock conditions. The action "THEN" prescribes the inhibition action of the entrance of new processes in the cycle (Fig. 6b). The pre-deadlock condition depends on the information of the CR states in real time, i. e., which resources are being used by which processes. The algorithm uses an adjacency list of RAG (ForwardStar and ReverseStar) to identify the processes, which belong to CWL and the respective entrances of these processes in the cycle. The figure 5c presents the algorithm to deduce the additional control rules.

4. CONCLUDING REMARKS

The main objective of this work was to introduce the new data structures to represents a RAG model and the respective algorithms for determination of CWL. The great advantage in use the adjacency list is regarding the reduction of the time to algorithm execution. The algorithm is based on depth-first search in adjacency list representation of RAG. The generation of resources allocation rules to avoid deadlock situation in FMS with multiple instances of resource. Basically, the flow-in-suppression deadlock avoidance policies are adopted. The next stage of this works is the implementation of computational tools using these data structures and the algorithms. Another important aspect is regarding the effect of the control rules in consecutive cycles.

$$Q_c - P_c = 1$$

a)



b)

$$IF\,(Q_c - P_c = 1)\,THEN\,(\mathrm{a}_{\mathrm{IN2}} = \text{false and } \mathrm{a}_{\mathrm{IN1}} = \text{false})$$

```
DeduceRules()
For (each CWL in the list of CWL) do
    Create the rules IF (pre-deadlock
    condition of CWL) THEN (inhibit the
    entrance of new processes in the cycle)
END
```

c)

Figure 6 The pre-deadlock condition (a), example of control rules (b) and the algorithm to deduce the rules (c).

5. ACKNOWLEDGEMENTS

The Authors would like to thank CAPES, CNPQ and CUFSA for their support to this work.

6. REFERENCES

Banaszak, Z.A. and Krogh, B.H., 1990, “Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows”, in IEEE Transctions on Robotics and Automation, vol.6, pp.724-734.

Banaszak, Z.A. and Polak, M., 2002, “Deadlock-free Distributed Control for Repetitive Flows”, Proceedings of the Sixth International Workshop on Discrete Event Systems, pp.273-278.

Calinescu, A., Efsathiou, J., Sivadan, S., Schirn, J., Huaccho Huatuci, L., 2000, “Complexity in Manufaturing: An Information Theoretic Approach”, in Proceedings of the International Conference on Complex Systems and Complexity in Manufacturing, Warwick University, Warwick.

Cassandras, C.G., 1993, “Discrete Event Systems – Modeling and Performance Analysis”, Richard D. Irvin, Inc. and Aksen Associetes, Incorporated Publichers, pp.1-139.

Cho, H., 1993, “An Intelligent Workstation Controller for Computer Integrated Manufacturing”, Doctor dissertation, Texas A&M University, pp.83-88.

Cormen, T.H., Leiserson, C.E., Rivest, E.L., Stein, C. 2003, “Introduction to Algorithms”, Second Edition, McGraw-Hill.

Drozdek, A. 2002, “Estrutura de Dados e Algoritmos em C++”, Editora Pioneira Thomson Learning.

Edmonds, B., 1995, “What is Complexity? – The Philosophy of Complexity per se with Application to some Examples in Evolution”, Symposium: The Evolution of Complexity, University of Brussels, Breussels, Belgium.

Ezpeleta, J., Trincas, F., Garcia Valles, F., Colom, J.M., 2002, “A Banker’s Solution for Deadlock Avoidance in FMS with Flexible Routing and Multiresource States”, IEEE Transactions on Robotics and Automation, vol. 18, pp. 621-625.

Ezpeleta, J., Recalde, L., 2004, “A Deadlock Avoidance Approach for Nonsequential Resource Allocation Systems”, IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 34, pp.93-101.

Fanti, M.P., Maione, B., Mascolo, S., Turchiano, B., 1997, “Event-Bases Feedback Control for Deadlock Avoidance in Flexible Production Systems”, in IEEE Transaction on Robotics and Automation, vol. 13, no.3, pp.347-363.

Fanti, M.P., Maione, B., Mascolo, S., Turchiano, B., 2000, “Comparing Digraph and Petri Net Approaches to Deadlock Avoidance in FMS”, in IEEE Transaction on Systems, Man, and Cybernetics, vol. 30, no.5, pp.783-798.

Hsieh, F.S., 2000, “Reconfigurable Fault Tolerant Deadlock Avoidance Controller Sysnthesis for Assembly Production Processes”, IEEE International Conference on Systems, Man, and Cybernetics, vol.4, pp. 3045-3050.

Hsieh, F.S., 2002, “Modeling and Control of Holonic Manufacturing Systems Based on Extended Contract Net Protocol”, Proceeding of the American Control Conference, vol.6, pp.5037-5042.

Ho, Y.C. and Cao, X.R. 1991, “Pertubation Analysis of Discrete Event Dynamic Systems”, Kluwer Academics Publisherds.

Ito, Y., 1991, “A Diserable Production Structure Lookin Toward the 21th Century – Anthropocentric Intelligence-Bases Manufacturing”, XI Congresso Brasileiro de Engenharia Mecânica, São Paulo, Brazil, pp.23-32.

Kumaran, T.K., Chang, W., Cho, H., Wysk, R.A., 1994, “A Structured Approach to Deadlock Detection, Avoidance and Resolution in Flexible Manufacturing Systems”, International journal of production research, vol.32, no.10, pp. 2361-2379.

Lewis, F.L., Gurel, A., Bogdan, S., Doganalp, A., Pastravanu, O.C., 1998, "Analysis of Deadlock and Circular Waits Using a Matrix Model for Flexible Manufacturing Systems", Automatica – Elsevier Science, vol. 34, No. 9, pp. 1083-1100.

Miyagi, P.E., 1996, "Controle Programável – Fundamentos do Controle de Sistemas a Eventos Discretos", Ed.Edgard Blucher Ltda, São Paulo, Brazil, pp.1-5, 71-148.

Murata, T., 1998, "Petri Nets: Properies, Analysis and Applications", in Procedings of the IEEE, vol.77,no4, pp.541-580.

Nakamoto, F.Y., Maruyama, N., Miyagi, P.E., Santos Filho, D.J., 2001ª, "Regras de Controle para Alocação de Recursos em Sistemas Produtivos com Processos Concorrentes", V Simpósio Brasileiro de Automacao Inteligente, Canelas, R.S., Brazil.

Nakamoto, F.Y., Maruyama, N., Miyagi, P.E., Santos Filho, D.J., 2001b, "Geração Automatica de Regras Adicionais de Controle para Alocação de Recursos em Sistemas Produtivos com Processos Concorrentes", XVI Congresso Brasileiro de Engenharia Mecânica, Uberlândia, MG, Brazil.

Nakamoto, F.Y., Miyagi, P.E., Santos Filho, D.J., 2002a, "Uma Proposta de Algoritmo para a Determinação dos Ciclos Fechados de Espera em Sistemas Produtivos Flexíveis", XVI Congresso Brasileiro de Automática, Natal, RN, Brazil.

Nakamoto, F.Y., 2002b, "Sistematização do Projeto do Controle de Sistemas Produtivos", Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil. (In Portuguese)

Nakamoto, F.Y.; Miyagi, P.E.; dos Santos Filho, D.J., 2003, "Systematization of the Project of the Production System Control", IEEE International Symposium on Industrial Electronics, Vol. 2, pp. 868-873.

Palazzo, L.A.M., Castilho, J.M.V., 1998, "Sistemas Complexos e Auto-Organizacao", VI Escola Regional de Informática, Curitiba, PR, Brazil. (In Portuguese)

Peterson, J.L., 1981, "Petri Net Theory and the Modeling of Systems", Prentice-Hall.

Reisig, W., 1985, "Petri Nets, An Introduction", Springer-Verlag, Berlin Heidelberg.

Santos Filho, D.J., Miyagi, P.E., 1995, "Enhanced Mark Flow Graph to Control Flexible Manufacturing Systems", Journal of The Brazilian Society of Mechanical Sciences, ABCM, Rio de Janeiro, 17(2):232-248.

Santos Filho, D.J., 2000a, "Aspectos do Projeto de Sistemas Produtivos", Tese de Livre Docência, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil. (In Portuguese)

Santos Filho, D.J., Motohashi Matsusaki, C.T., Nelli Silva, E.C., Miyagi, P.E., 2000b, "Automatic Generation of Metamodels to Design Production Systems Control", in Proceeding of Mechatronics 2000. 7th Mechatronics Forum International Conference, Atlanta.

Stecke, K. E., 1986, "A Hierarchical Approach to Solving Machine Groupinf and Loading Problems of Flexible Manufacturing Systems", European Journal of Operational Research, Elsevier Science Publishers B.V., North-Holland, pp.369-378.

Viswanadham, N., Narahari, Y., Johnson, T.L., 1990, "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models", in IEEE Transactions on Robotics and Automation, vol.6, no6, pp.713-723.

7. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.