16

EFFICIENT EVENT HANDLING IN SUPPLY NETWORKS USING Q-LEARNING AND K-MEANS CLUSTERING

Andre Doering
Fraunhofer ALB Paderborn, andre.doering@alb.fraunhofer.de
Wilhelm Dangelmaier
Heinz Nixdorf Institute, University of Paderborn, whd@hni.upb.de
Christoph Laroque
Heinz Nixdorf Institute, University of Paderborn, laro@hni.upb.de

Modern value-added processes will be globally cross-linked through outsourcing and reduction of real net output ratio. Therefore logistical planning and control processes become more complex. Events in supply networks and their consequences to the partners in the supply network will be hardly to overlook without using computer based decision support systems. This paper describes such a decision support system, learning the rules used to control the production network. In details the system architecture will be described, requirements to such a system will be identified and a solution developed at the Heinz Nixdorf Institute and Fraunhofer ALB (application center for logistic-oriented business administration) in Paderborn will be presented. The solution is based on a q-learning approach supported by a k-means clustering algorithm.

1. INTRODUCTION

Modern value-added processes in the European Automotive Industry will be globally cross-linked through outsourcing and reduction of real net output ratio to reduce production costs (Fraunhofer, 2004). Therefore logistical planning and control processes become more complex, because more cross-linked processes cause higher co-ordination effort for planning and controlling such heterogeneous supply networks (Baumgaertel, 2006).

Especially the handling of events, causing direct effects to the supply network and its production systems, must be handled very efficient. To optimize such control processes in their efficiency and reliability, automated systems are used to support human production planners in their daily complex decisions making processes. But the complexity of the event handling task in supply networks limits the usage of classical operation research methods and their algorithms: adequate models to model the problem will cause NP-hard optimization problems and long algorithm runtimes (Suhl, 2006).

The handling of an event needs fast reaction, at best case in real-time (Doering, 2007). Therefore, the usage of *intelligent* methods for production network control like artificial learning systems is in the focus in applied production research both in applied scientific projects (AC-DC) and in industry (Diedrichsen, 2007). Despite their specific

implementation the objective of such intelligent systems is mostly to learn rules supporting human or automatic event handling by selecting possible reaction measures.

An event is defined as a state of a production plan offering a restriction violation in this production plan after an unexpected change of customer demands, suppliers or capacity supply or demand change, e.g. usage of safety stock after an increased customer demand. Reactions to events are here defined as the usage of specific change panning strategies, implemented by specific change planning algorithms for solving occurring lacks in production plans efficiently (Heidenreich, 2006).

Event-based rules have been defined to select applicable planning strategies enabling a fast reaction to the event (Ibid.). But the complexity of the supply networks generates many possible event states, which requires many rules to cover all possible or relevant event situations. It is obvious, that a human planner is not able to formulate all rules for implementing an efficient rule based event handling system. Also the usage of experience causes problems, because experience may not cover all respectively future events and is hardly to extract objectively using knowledge engineering techniques (Görz, 2003).

Therefore, this work deals with using machine learning techniques based on Q-learning (e.g. Mitchell, 1997 or Sutton, 1998) to learn such rules automatically and though efficiently.

For the implementation of such a learning system, several tasks are to be fulfilled. The complexity of the state space is a problem causing nearly unlimited exploration times for the Q-learning algorithm. Moreover, the learning function for Q-learning reward calculation must be defined, regarding the objective of a supply network based learning task. Thirdly, an efficient training algorithm must be developed to train the learning system efficiently.

This paper will focus on the definition of such a learning system and outline a concept for state reduction and the calculation of rewards. Forthcoming problems for training and testing will be discussed briefly.

The paper will start with brief definition of the learning problem and the requirements to the learning system in detail in chapter. 2. Chapter 3 outlines an overview of the state of the art. Chapter 4 introduces the concepts and drafts first results. The paper closes with a summary of the achieved work and an outlook to forthcoming research activities.

2. PROBLEM DEFINITION

2.1. Co-ordination in supply networks

The basic character of an automotive supply network is the breakdown of value adding processes to several production stages staring at *tier-n* up the Original Manufacturer (OEM), which finishes the value adding process by the final assembly of the car. Most production stages are based on serial production, which is mainly planned by lot scheduling algorithms (Heidenreich, 2006).

In the production system model language MFERT (Schneider, 1996) (see Figure 1) every stage of a supply network can be modeled as combination of capacities (CON)[1], processes (PN)[2] and an incoming edge for material out of a buffer (AON)[3], e.g. stock. The

[1] Capacity object node

[2] Process node

[3] Assembly Object Node

stages are connected between an edge from a PN to an AON.

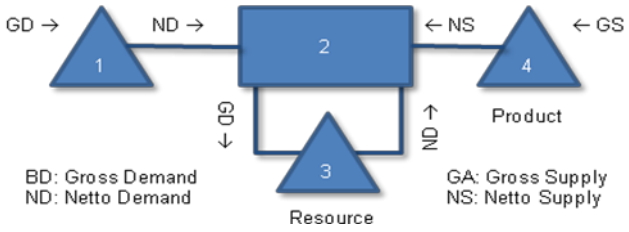

Figure 1. Basic MFERT Model

Considering a network consisting of several connected stages, the material flow generally starts from the tier-n up to the OEM. The information flow, consting of demands for material in a certain time period, is directed upwards and downwards the network, depending on the specific type of information exchange. Upward flowing information is called supply[4]. It contains information about part supplies based on values, that will be procured to the earliest possible period of a production pan in the next stage. Downward information flow will consist of demand figures representing a latest point in time for delivering a part to the production stage, where the demand is generated.

A plan consists of a number of periods t starting from now (t_0) until a pre-defined last period $PH : t \in 1,\ldots,PH$. Every period of a plan is, depending whether it is an AON or CON, allocated with a stock value or capacity value called lots. Every plan for an AON/CON will generally be represented by a vector p∗ allocated with a lot to consisting periods of a plan denoted by vector $p^*(t)$.

Every period will be restricted by a maximum and minimum value representing the maximum space of a buffer respectively the minimum safety stock or the maximal utilization of a modeled capacity. Between every production stage the flow of material is limited to a min or max value. Additionally, the upward procurement process between production stages can be managed in recurring cycles or at any point in time.

If a min/max-restriction is violated, the corresponding plan is called 'not consistent' and a change planning process has to be started, in order to generate a new consistent plan. The coordination during those change planning processes in the supply network is organized in a decentral way. In the implementation of the learning system, every production stage is an autonomous agent, that coordinates only by communication with the prior or succeeding production stage agents by sending gross or supply figures. For every type of change planning coordination a specific planning algorithm has been classified, that reschedules a local plan or globally sends requests of demands or supplies to corresponding agents of other production stages by demanding a minimum of plan changing steps to finish the rescheduling process.

Human defined rules are used to choose a change planning algorithm, based on certain feasible system states to assure a fast recovery to a consistent plan without the need for many planning cycles.

Such a system consists of an exponential growth of states, depending on the size of the network, the number of planning horizons and the min/max-restriction of every period. So the definition of control rules, based only on state/action-pairs, will lead to a tremendous number of rules and cannot be generated e.g. by a human planner anymore.

[4] [4]Supply can be split into *netto*(NS) and *gross*(GS) supply depending on the point in the MFERT model where it is consumed by a node (see figure 1)

2.2. Requirements for a production control rule learning system

Wanting to be able to learn those rules in such an enormous state space, several problems arise. At first, the learning system must be able to search the state space efficiently and reduce its runtime to an acceptable time period. Secondly, the learning task itself must be based on an intuitive learning function, since the usage as a decision support system, the acceptance of the learned rules will rely on their intuitive understanding by human planners. At last, the training process of such learning system learning on a distributed and interrelated network planning model must be well defined to prevent the extension of runtime duration through never-ending planning processes.

Therefore, Q-Learning is a suitable solution for learning rules, because the reward function, which fulfills the learning task, can be specified in a problem oriented and intuitive way. For convergence, Q-Learning must search an unlimited amount of time through the state space (Mitchell, 1997). Therefore the state space must be reduced, in order to make Q-Learning efficient while assuring convergence of the Q-values.

Clustering, especially k-means clustering function has been identified in former and is described in a published paper, using a problem-oriented clustering (Doering, 2007 p. 487-497).

The training process must combine both, clustering and Q-learning, to an efficient learning system. The training process should deal with learning episodes based on a change planning negotiation process and restricted by clear stopping rules. This requirement prevents the unlimited duration of a learning episode and the learning task itself. The whole training process will coordinate the learning episodes and stops the training, when successful.

moreover, the quality of the original data, used in the learning process is of a high importance. Only real data, e.g. extracted from ERP-Systems, or realistic generated data must be used to ensure a problem oriented learning task.

In general the learning task could be described as:

> *Learning of production system control rules will be enabled by a problem oriented and intuitive mathematical assessment of change planning actions (reward). The rules will be represented by sorted list of Q-values where every Q-value represents a proposal for a suitable change planning action based in a specific state. The training process must rely on problem-specific real or realistic original data to make the learning process must efficient.*

In sum, the core question for a learning task is illustrated in Figure 2. Can an event be handled locally e.g by reducing safety stock (1), or should the gross demand of supplier 1 (2) or supplier 2 (3) be reduced to handle this event. Every learned rule will propose a solution for such a question.

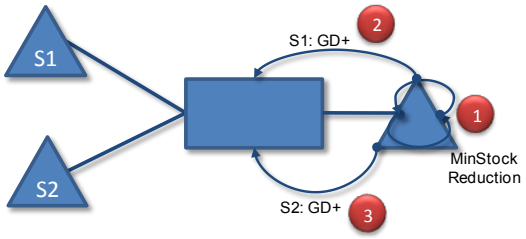

Figure 2. Draft of the learning problem

Therefore, the next chapter will shortly summarize the state of the art in Q-learing of production rules. The clustering part is also covered by (Döring, 2007 p 487-497).

3. STATE OF THE ART

In general, only few applications in Q-Learning deal with learning in distributed network models. Stegherr (Stegherr, 2000) developed a Q-Learning approach, used for control of anticipated job control in production systems. This system is based on learning local decision and therefore not suitable for this learning task.

Stockheim et. al. (Stockheim, 2003) conceptualized a learning approach for supply chain management. The learning task is to plan local lots for production charges and generate from this a secondary demand for the next production stage. The production system model used in this work is not sufficient for this learning task, because of its high granularity.

Cao et. al. focused on learning fabrication fulfillment figures for a 2-stage production system not equivalent to a supply network. Mahadevan et. al. developed a learning concept called SMART learning rules for machine maintenance in factories. This concept could be used in addition to a change planning learning approach, but is actually not sufficient for usage in this work.

4. LEARNING PRODUCTION CHANGE PLANNING CONTROL RULES

For learning control rules using Q-Learning support by *k-means* clustering the architecture proposed in Figure 3 is used.

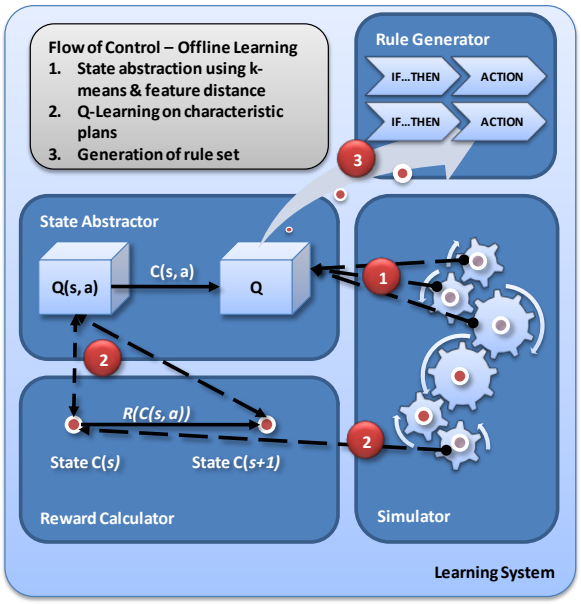

Figure 3. Learning System Architecture

The architecture consists of four modules. The *State Abstractor* module implements the clustering methods used to reduce the state space efficiently (Döring, 2007 p. 487-497). The Q-values are assigned to admit state/action pairs on cluster level, while each cluster is represented by a characteristic state. This state, called Centroid, represents the characteristic course of all assigned single states.

The R*eward Calculator* module assesses the state/action-pairs during the training process and calculates the Q-values. Each observed state will be mapped to its cluster, the Centroid is taken as origin data for the event handling change planning process. Then the resulting state is again mapped to its cluster and the reward between *C(s,a)* and *C(s+1)* will be calculated and the Q-value of *C(s)* will be updated.

The *Simulator* module provides the learning system with original data based on real data or realistically generated data. Furthermore, this module controls the training process and its learning episodes.

The *Rule Generator* Module generates, based on a specific algorithm, the rules from the preordered Q-values of each clusters stat/action pair.

4.1 Concept of the reward calculation function

To calculate the reward in an intuitive way, factors used for the decision taking by production planner should be considered. Based on the general approach of assessments in economics, a cost function will be used as a basis for the reward calculation.

Main costs factors in production networks are that are regarded for assessment of plans are (Gudehus, 2004):

- *Preparation costs:* In AON materials must be provided for transformation in the production process. This could be assessed by this preparation costs eg. including fix costs (stock etc.) and variable costs (e.g. employees).
- *Procurement costs:* If material is not available from stock, it will be procured from suppliers respectively every procurement process causes costs. These costs are based on specific agreement between a supplier and a customer based e.g. on the value and regularity of parts that are procured.
- *Resource costs:* To transform material resources, e.g. machine capacity, is needed. The performed work can be assessed by costs.
- *Restriction violations:* Every plan restriction violation, namely an event, causes overhead for its handling e.g. through the demand of troubleshooters, who deal with those topics in their daily works. This overhead can be represented by costs.

A plan will be assessed based on its periods and their values and restrictions. To get a normalized reward value the assignment of plan periods costs are normalized and limited to the interval *[0..1],* while *0* represents no costs and *1* maximal costs caused by restriction violations in one period.

Based in the assumption, that events occurring in nearer future have more impact than events occurring later the costs will be reduced to the end of the planning horizon by a discount factor *DF(p(k)):*

$$DF(p(k)) = \left(\frac{1}{1+discount}\right)^{k}, k = 1..PH, p(k) \in P, discount \in 0..1$$

The sum of the costs of all periods in a plan is equal to the costs caused by this plan, called penalty costs *PC(P)* in state *s*.

$$PC(P_s) = \sum_{k=1}^{PH} DF * PC(p(k))$$

Based in this the reward can be generally defined as the difference between the penalty costs in state *s* and state *s+1* after a processed change planning.

$$R(P_s, a_i) = PC(P_s, a_i) - PC(P_{s+1})$$

The main task is to calculate the detailed penalty costs of each period as the basis for the reward calculation. Despite the differences between global and local change planning Figure 4 shows the general concepts that is used.

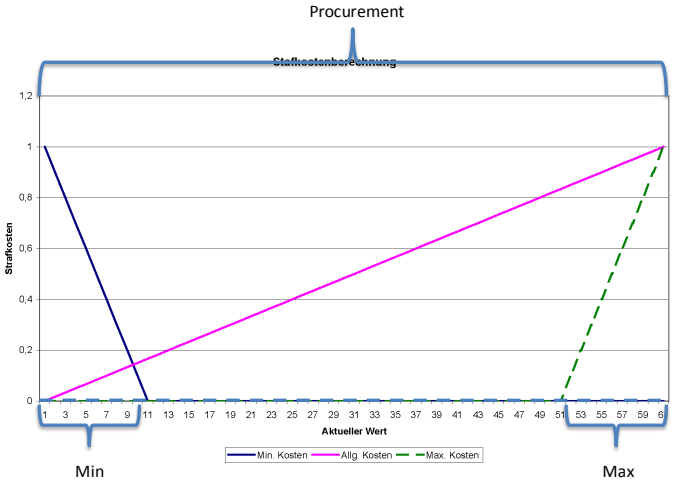

Figure 4. Penalty cost concept

A period is restricted by a minimal and maximal restrictions while values above or below this restriction cause a restriction violation and an event. Therefore this could be assessed by a specific cost function. Also general costs for procurement occur in each period raising to the value of material procured. This could also be assessed by a cost function. The full penalty costs can then be calculated from the sum of the specific cost type in each period. To get comparable costs those cost types are normalized to an interval between *[0..1]*.

5. SUMMARY

This paper discussed the requirements for a rule learning system to control change planning processes in production networks. Learning system architecture has been introduced, based on k-means clustering and Q-Learning.

The general approach for calculation of rewards based on cost functions and the storage of the Q-values have been drafted out.

Further work will specify the detailed cost functions for local and global change planning processes and detail the training process. An implementation of the system will be carried out to validate the learning architecture and the effects of clustering to its convergence.

6. REFERENCES

1. AC-DC Automotive Chassis Development for 5Day-Cars. European Integrated Research Project Sixth Framework Program. Contract No. 031520. http://www.acdc-project.org
2. Baumgaertel H.; Hellingrath B.; Holweg M., Bischoff J. Automotive SCM in einem vollständigen Build-to-Order-System. Supply Chain Management. 2006; 1: 7-15.
3. Cao H, Smith SF. "A Reinforcement Learning Approach to Production Planning in the Fabrication/Fulfillment Manufacturing Process". In Proceedings of the Winter Simulation Conference, Chick s, Sanchez PJ, Ferrin D, Morrice DJ. 2003.
4. Diedrichsen K, Nickerl, RJ. Interview: Intelligenter als das reine Event. Logistik Heute. 2007; December. 16-18.
5. Doering A, Dangelmaier W, Danne C. "Using k-means for clustering in complex automotive production systems to support a Q-learning-system". ICCI In Proceedings of the 6th IEEE International Conference on Cognitive Informatics Zhang D, Wang Y, Kinsner W (ed.). 2007; 487-497.
6. Doering A, Dangelmaier, W, Laroque C, Timm T. "Simulation-aided process coverage for delivery schedules under short delivery schedules using real-time event based feedback loops". In Proceedings of the 6th EUROSIM Congress on Modelling and Simulation. 2007; Vol 1.
7. Fraunhofer Gesellschaft, Mercer Management Consultants. Future Automotive Industry (FAST) 2015. Mercer Management Consultants. 2004.
8. Goerz G, Rollinger C, Schneeberger J (ed.). Handbuch der künstlichen Intelligenz. Oldenbourg Wisssenschaftsverlag. 2003.
9. Gudehus T. Logistik. Springer Berlin Heidelberg. 2004.
10. Heidenreich J. Adaptive Mengen- und Kapazitätsplanung in kollaborativen Produktionsnetzwerken der Serienfertigung. Dissertation. University of Paderborn. 2006.
11. Mahadevan S, Marchalleck N, Das TK, Gosavo A. Self-improving Factory Simulation using Continuous-time Average-Reward Reinforcement Learning. Techreport IRI-9501852. Department of Computer Science and Engineering University Of Floriada. 1997.
12. Mitchell TM. Machine Learning. McGraw-Hill Bool Co. 1997.
13. Schneider U. Ein formales Modell und eine Klassifikation für die Fertigungssteuerung. Dissertation. Universität GH-Paderborn. 1996.
14. Stegherr T. Reinforcement-Learning zur dispositiven Auftragssteuerung in der Variantenreihenproduktion. Herbert Utz Verlag. 2000.
15. Stockheim T, Schnwind M, Koenig W. "A reinforcement learning approach fior supply chain management". In 1st European Workshop on Multi-Agent Systems. 2003.
16. Suhl, L. Mellouli T. Optimierungssysteme. Modelle, Verfahren, Software, Anwendungen. Springer. 2006.
17. Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Bradford Book – MIT Press. 1998.