Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

An Autonomous Control Concept for Production Logistics

Henning Rekersbrink, Bernd Scholz-Reiter, Christian Zabel

BIBA - Bremer Institut für Produktion und Logistik GmbH, Hochschulring 20, 28359 Bremen, Germany
{rek, bsr, zbl}@biba.uni-bremen.de

Abstract. The German Collaborative Research Centre 637 'Autonomous Cooperating Logistic Processes' tries to make a paradigm shift from central planning to autonomous control in the field of logistics. Among other things, autonomous routing algorithms based on internet routing protocols are developed. The Distributed Logistics Routing Protocol (DLRP) was originally designed for transport networks to match goods and vehicles and to continuously make route decisions. Now the protocol was transferred to production logistics as a promising autonomous control method. The DLRP enables the abilities for logistic objects, orders and machines, to make own decisions with the information actually and locally available. In contrast to common scheduling algorithms, the DLRP is not a planning, but a control method with the capability for multiple, user defined optimization goals. The new autonomous control concept for production logistics will be presented in this paper and a first evaluation with common scheduling heuristics will be given.

Keywords: Flexible Flowshop, Scheduling, Dynamics, Autonomous Control

1 Introduction

Due to growing dynamics and complexity of logistics systems, common concepts of hierarchical planning and control are questioned. A possible alternative is a shift from central planning to decentralized, autonomous control strategies (see e.g. [1], [2], [3], [4]). This concept of autonomous control is the main research area of the German Collaborative Research Centre 637 „Autonomous Cooperating Logistic Processes – A Paradigm Shift and its Limitations".

One possibility to implement an autonomous control strategy is to transfer existing routing protocols from data communication to similar routing problems in transport logistics. This idea lead to the development of a new autonomous control concept called *Distributed Logistics Routing Protocol* (DLRP). The DLRP was originally designed for transport logistics. The general idea was to make a paradigm shift from central planning to decentralized, autonomous control where each logistic entity is able to interact and decide autonomously (see [5], [6], [7]).

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

After the suitability and performance of the DLRP concept was shown for the field of transport logistics ([6]), it was obvious to transfer the basic concept to other logistic fields, like production control. Due to this transfer, two different names are introduced: DLRPt and DLRPp for transport and production respectively. The long term vision is here to connect all concepts to one system where a product order is able to route its way through a production environment, between production sites (global production networks) and after its completion through a transport environment to reach the costumer at the right time.

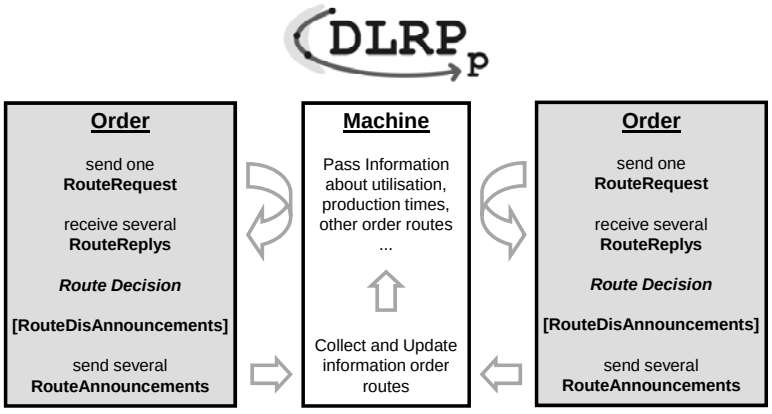

Fig. 1. Basic scheme for the Distributed Logistics Routing Protocol for Production

A brief overview about the DLRPp can be illustrated on the basis of fig. 1. When an order enters the system, it needs a route through the production environment. It sends a RouteRequest object to the next machine which fills it with necessary information. The machine now manifolds the RouteRequest object and sends ahead the objects to successional machines (not to all possible machines; this point is discussed later). These do the same until the last production step is reached. By this scheme each of the many RouteRequest objects represents a possible routes through the production set. At the last production step all the collected information are sent back to the order as a RouteReply object. This RouteReply object is basically a RouteRequest object on its way back to the order. The order receives several RouteReply objects as route alternatives by this discovery scheme. After its route decision, the order announces the new routes and possibly disannounces old routes from previous routing processes by sending RouteAnnouncement and RouteDisAnnouncement objects respectively. Together with the RouteAnnouncement, plenty of information can be passed to the relevant machines, like production times, setup times, probabilities, urgencies etc. Because of the ongoing processes, this scheme leads to a continuous cooperative structure - at any time there is enough information for any decision. Orders decide their preferred routes, machines decide their setup plan and the dispatching.

In contrast to classical scheduling algorithms, the DLRPp is designed for controlling an ongoing process. It does not need all information in advance – although

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

it would be possible to announce roughly planned future orders to the system. The structure of the DLRP leads to some crucial advantages like adaptivity, robustness, possible manual interventions, estimation of future net states and an arbitrary decision process (see [7]).

In the next section, the DLRPp concept is presented in detail. Section 3 then gives a brief evaluation of the DLRPp by taking common scheduling algorithms as reference. The problem used for this evaluation is a dynamic flexible flowshop problem with unrelated parallel machines and sequence-dependent setup times. The paper closes with a conclusion of the evaluation and an outlook on further research and application possibilities.

2 Distributed Logistics Routing Protocol for Production

The core function of this autonomous control concept is to route orders through a production logistics scenario. Figure 2 exemplarily shows a simple flexible flowshop scenario. At this, some optimization criteria may be defined such as minimization of the throughput time or the maximization of the machine utilization.

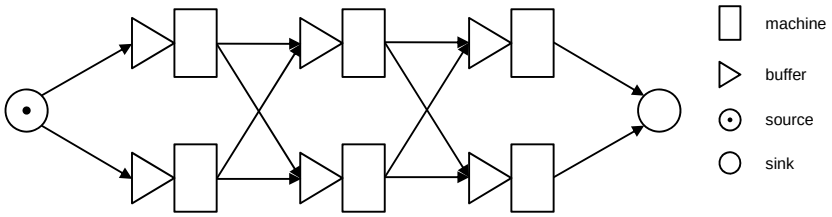

Fig. 2.: Example of a production logistics scenario

In order to deal with all possible optimization criteria, all logistic objects involved have to be defined within the protocol. For this case, the DLRPp defines two logistic object types: machines and orders. The main attributes for these objects are shown in Table 1, where italic names are representatives for larger data structures.

In addition to logistic objects, there are four different data objects defined by the DLRP: RouteRequest, RouteReply, RouteAnnouncement and RouteDisAnnouncement. These objects are data packages, which are sent from one logistic object to another. Every single instance of these data objects belongs to one routing process of one order. To identify this process, every data object has the two attributes ‘OwnerID’ and ’RouteRequestID’ according to the order attributes. The routing information itself is stored within these data objects as a special data structure, which may be user-defined according to the application.

To give a detailed description of the DLRPp, let’s follow an order object through the system. Figures 3 and 4 show the procedures of the logistic objects, which are described in the following. Firstly, the order has to be initialized. This means that the order object does its first routing process. It gets a ‘StartRouting’-message from the

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

system (see fig. 3). The order is in 'no Routing'-state, so it creates a new RouteRequest object and sends it to all possible machines for the first production step (this information has to be defined within the 'Production Steps'-attribute). After that, it sends two delayed messages to itself. One is to ensure the next routing process after a maximum time interval (MaxRoutingInterval). The other one is to ensure the end of the actual routing process after a maximum routing time (MaxRoutingTime) – even without any answer from the system. The 'WatchRouting'-process can be retraced with fig. 3.

Table 1.: Main attributes of the logistic objects

Logistic Object: Order	
Name	*Type*
ID	string
Type	string
ReleaseTime	double
DueTime	double
Production Steps	
MaxRoutingInterval	double
MaxRoutingTime	double
MaxCountReceivedRouteReplys	integer
MaxAnnouncedRoutes	integer
Position	
State ('routing'/'no routing')	string
RouteRequestID	integer
CountReceivedRouteReplys	integer
Data from routing process	
...	

Logistic Object: Machine	
Name	*Type*
ID	string
OrderTypes	strings
RetoolingTimes	double
ProcessingTimes	double
TransportTimes	double
Neighbour-IDs	strings
Data for buffer content	
Data for announced orders	
Data for server	
...	

The RouteRequest objects are received by a machine as a 'GetRouteRequest'-message (see fig. 4). The machine fills up the RouteRequest with relevant data such as expected waiting time or processing time for the order type. This point (see '1' in fig. 4) is discussed below. The crucial point here is, that the machine writes its own ID, expected arrival and departure time into a list of the RouteRequest. By this procedure, the way of a RouteRequest object is collected and creates a possible route for the order.

If this machine is not the last production step for the order, it generates a recipient list of possible following machines for the order (this information must be placed within the RouteRequest). For the performance of the protocol, this is a very crucial point (see '2' in fig. 4) and is discussed below. Multiple copies are made of the RouteRequest object and they are sent to all recipients. These recipients follow the same scheme. When a machine is the last production step, it sends back the RouteRequest as a RouteReply to the order. This RouteReply object has now all data from all machines on its way.

The order now receives the RouteReply object as a 'GetRouteReply'-message. If the routing process is still active (state = 'Routing'), the data from the RouteReply is saved and the counter is counted up. When the order has received a maximum number of RouteReplys (MaxCountRouteReplys), the order sends an 'EndRouting'-message to itself. With this 'EndRouting'-message, the RouteReuestID is counted up and the order decides for a route (see '3' in fig. 3). The pool of received RouteReplys defines all route alternatives for this decision. This point is described below. After the routing

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

decision, the order disannounces older route decisions and announces the actual routes at the corresponding machines by sending 'RouteDisAnnouncement'- and 'RouteAnnouncement'-objects.

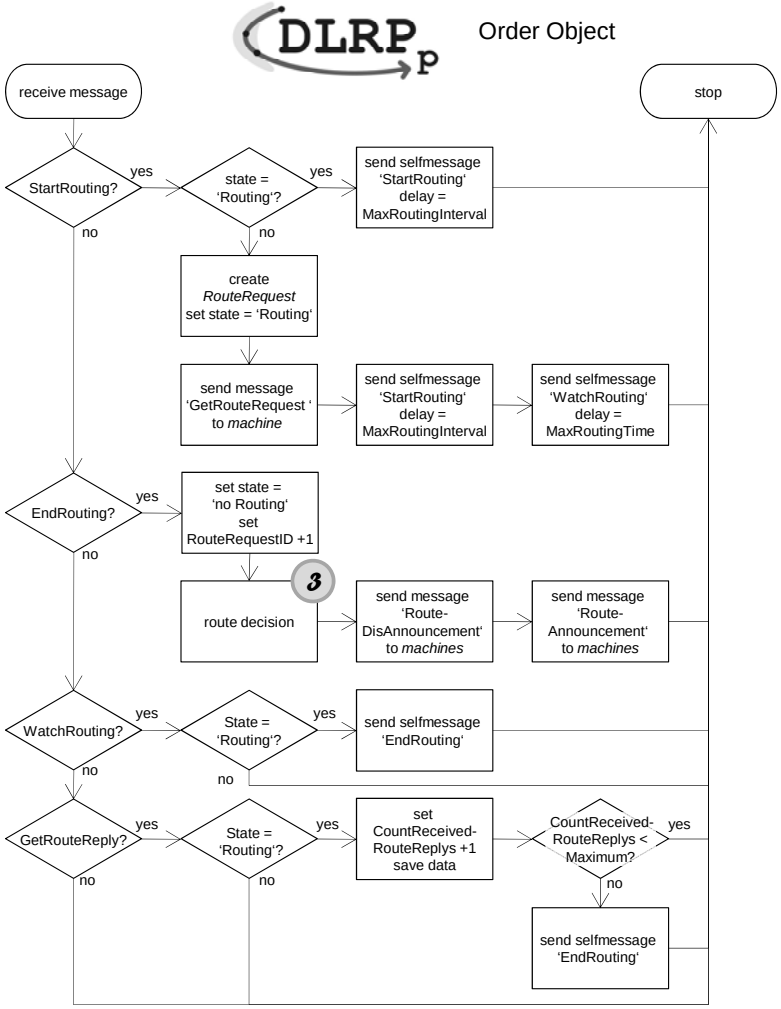

Fig. 3.: Chart for order objects

In the following, some important points and parameters for the autonomous control are described. There are many different possibilities to configure the DLRPp at these points. These possibilities are the focus of actual and future work concerning the protocol.

Route Decision. The decision process itself is not a part of the DLRP. It may be user defined depending on the application. The route decision of an order object is based on the received RouteReplys, so the machines have to put the necessary

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

information into the RouteRequest objects. For the evaluation presented below, the decision was made very simple for this first step. The smallest expected completion time C'_k is the only criterion for the decision. The completion time of an order is directly connected to its throughput time and the makespan (see below).

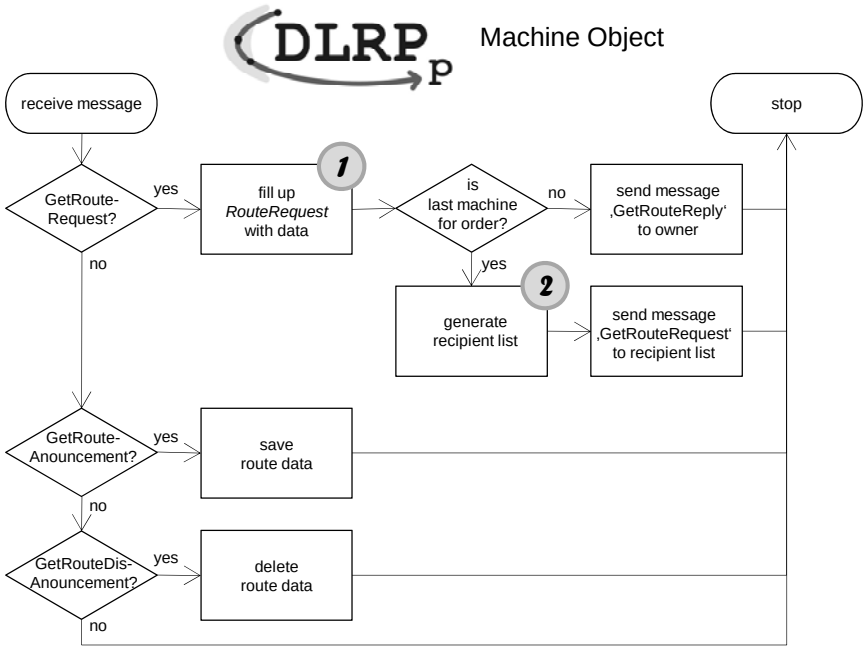

Fig. 4.: Chart for machine objects

MaxAnnouncedRoutes. The orders do not announce one definite but several possible routes. For these alternatives, there are values defined for the preference pr_k for the route k. The preference is defined by the expected completion time C'_k:

$$pr_k = \frac{1/C'_k}{\sum_k 1/C'_k}, \text{ where } \sum_k pr_k = 1$$

Fill up RouteRequest. According to the route decision criteria, the values for the RouteRequest are the arrival time, the estimated waiting time, the production time and finally the estimated departure time. The waiting time is estimated on the basis of the actual orders in the machine buffer and the announced orders. The workload from the announced routes are allocated with the preference of the route.

MaxCountRouteReplys. This value is important for the route decision quality. When this value is too small, there may be only bad RouteReplys as alternatives.

Generate Recipient List. This is a deciding point for the DLRP. In small scenarios, it is the best to sent the RouteRequest to every possible next machine. But this would lead to a combinatory explosion. Therefore, an intelligent method for the reduction of the recipient list is needed. For the actual DLRP, this reduction is realized in a complex way and is still under investigation.

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

Order Path Decision. This decision process is not part of the DLRP. It may be user defined depending on the application. When an order is finished at a machine, it has to decide where to go next. Because of the announcement of several routes, there possibly are alternative machines as next step. This decision is made randomly, while the probability for the alternatives are set to the according preference of the route. At this point, it has to be ensured that all announced routes have a chance to be chosen. Otherwise the announcement of more than one route has no effect. The random decision is a simple possibility here and it also ensures that the route preference of an order has an effect. Other possibilities which include machine states, for example, are still under investigation.

Dispatching. This decision process is not part of the DLRP. It may be user defined depending on the application. The order with the shortest retooling time is taken as next order.

3 Evaluation

In order to evaluate this new autonomous control concept, a *dynamic flexible flowshop scenario with unrelated parallel machines* was chosen. Figure 2 sketches an exemplary scenario with three stages and two parallel machines. On the basis of this production scenario, a brief evaluation study for the DLRPp is presented below.

A crucial advantage of the DLRP is the ability to optimize multiple goals. The decision points can be user defined depending on the application. For a complete evaluation of the logistic performance of a system many key figures are relevant. Work in process, utilization, throughput time and due date reliability are some of these values (see e.g. [8]). The reference heuristics for this evaluation study try to minimise the makespan of the system as a single goal. In order to keep the comparability between the systems, the DLRP had to be simplified to minimise the makespan only.

3.1 Problem Description and Instances

In order to obtain comparability for this evaluation, an existing problem formulation from Jungwattanakit, Reodecha, Chaovalitwongse and Werner [9] is chosen. It defines a flexible flowshop problem with unrelated parallel machines and sequence-dependent setup times. The term dynamic denotes the nature of the job arrivals defined by Conway (see [10]). In a static problem a certain number of jobs arrive simultaneously. In a dynamic problem the shop is a continuous process - Jobs arrive intermittently. In addition to this problem formulation, order types were defined here. Order types assign orders into groups with equal production and setup times.

Consider a flexible flowshop system like sketched in fig. 2. There are T stages and M^t unrelated machines at every stage t. All J orders have to pass every stage from $t = 1$ to $t = T$, whereas the machine choice in every stage is free. There are S different order types defined for all orders. All order types have different processing times $p_{m,s}^t$ on the different machines and they have different sequence-dependent

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

setup times $s^t_{m,u,s}$. The completion time C_j and the throughput time T_j are defined for the whole network at the end of last stage $t = T$.

Parameters:

J	number of orders
S	number of order types (indices s and u)
T	number of stages
M^t	number of parallel machines at stage t
r_j	release time of order j
$s^t_{m,u,s}$	setup time from order type u to s at machine m at stage t
ps^t_s	standard processing time of order type s at stage t
v^t_m	relative speed of machine m at stage
$p^t_{m,s}$	processing time of order type s on machine m at stage t where $p^t_{m,s} = ps^t_s / v^t_m$

Variables:

C_j	completion time of order j
C_{max}	makespan, where $C_{max} = \max(C_j)$
T_j	throughput time of order j; where $T_j = C_j - r_j$
TPT	mean throughput time; where $TPT = mean(T_j)$

The problem instances for this evaluation were chosen very close to the so called 'large size problems' defined by Jungwattanakit et al in [9].

- the number of orders J is 250
- the number of order types S is set to 10, they are uniformly distributed
- the number of stages T and the number of parallel machines M^t are set to the variants 3x5, 5x3, 5x10 and 10x5
- the standard processing times ps^t_s are integers uniformly distributed in the interval [1 50]
- the relative speeds v^t_m are uniformly distributed in [0.7 1.3]
- the setup times $s^t_{m,u,s}$ are integers uniformly distributed in [1 10]
- the release times r_j are all 0 (st, static) or a poisson process with λ orders per time; λ can be st, 0.5, 0.1, 0.075, 0.05, 0.025 for the small and 0.5, 0.1, 0.075, 0.05 for the larger scenarios (due to computation time). Lambda can be considered as a parameter for the dynamic or the workload of the scenario.

3.2 Reference Heuristics

The flexible flowshop scheduling problem itself is NP-hard (see e. g. [11]). Therefore there are no algorithms known so far for finding an optimal solution for a larger *dynamic* flowshop scheduling problem in feasible time, not to mention unrelated parallel machines. In order to find approximate solutions, many heuristic methods were developed for the classical flexible flowshop problem (see [12], [13] or [14] for an overview for the more general job shop scheduling). These heuristics were taken as basis for the development of heuristics for the dynamic flowshop problem.

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

Jungwattanakit et al made large simulation studies to evaluate some heuristics for the flexible flowshop problem mentioned (see [9]). They used the following sequencing algorithms as basis for their heuristics:

- PAL: a slope index heuristic by Palmer, see [15]
- CDS: a best choice heuristic by Campbell, Dudek and Smith, see [16]
- GUP: a slope index heuristic by Gupta, see [17]
- DAN: a heuristic by Dannenbring, see [18]
- NEH: a constructive heuristic by Nawaz, Enscore and Ham, see [19]

All heuristics try to minimise the makespan of a given problem instance. They are taken as reference algorithms to evaluate the new developed method.

3.3 Results

The problem instances were solved by all five reference algorithms and by a DLRPp simulation. The results of these calculations are shown in figures 5 and 6. Each point in the graphs represents an average value for five instances. To have an estimation for a lower bound for the makespan, an average of a minimum makespan is calculated:

$$\min C_{max} = \max\begin{pmatrix} \max(r_j) + T \cdot \text{mean}(p^t_{m,s}) + T \cdot \text{mean}(s^t_{m,u,s}) \\ J \cdot (\text{mean}(p^t_{m,s}) + \text{mean}(s^t_{m,u,s}))/M^t \end{pmatrix}$$

Similar to the makespan, a lower bound for the throughput time is estimated:

$$\min \text{TPT} = T \cdot \text{mean}(p^t_{m,s}) + T \cdot \text{mean}(s^t_{m,u,s})$$

Concerning the makespan, fig. 5 shows that NEH is the best of the reference algorithms. For more static scenarios (larger λ or $\lambda = st$) and those with less stages than parallel machines, the DLRPp and the NEH are both close to the lower bound. One can say that they are more or less equal for these problem instances. For instances with more stages than parallel machines, the DLRPp gets better than the NEH heuristic only with more dynamic workload (smaller λ). This is quite reasonable, because the scheduling algorithms are originally made for static scenarios and they are best with smaller scenarios, because of the extraordinary increase of scheduling alternatives with larger scenarios. Around a reasonable system workload of $\lambda = 0.1$, the DLRPp gets better than the reference and it even gets near to the estimated minimum.

In fig. 6, one can see the main disadvantage of scheduling algorithms in a dynamic environment. They are made to optimize a schedule for a given set of orders, but they don not look on throughput times. For an ongoing production process, where orders continuously enter and leave the system, they cause high throughput times. The DLRPp in contrast is made for a continuous process - therefore the results are best for dynamic instances.

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

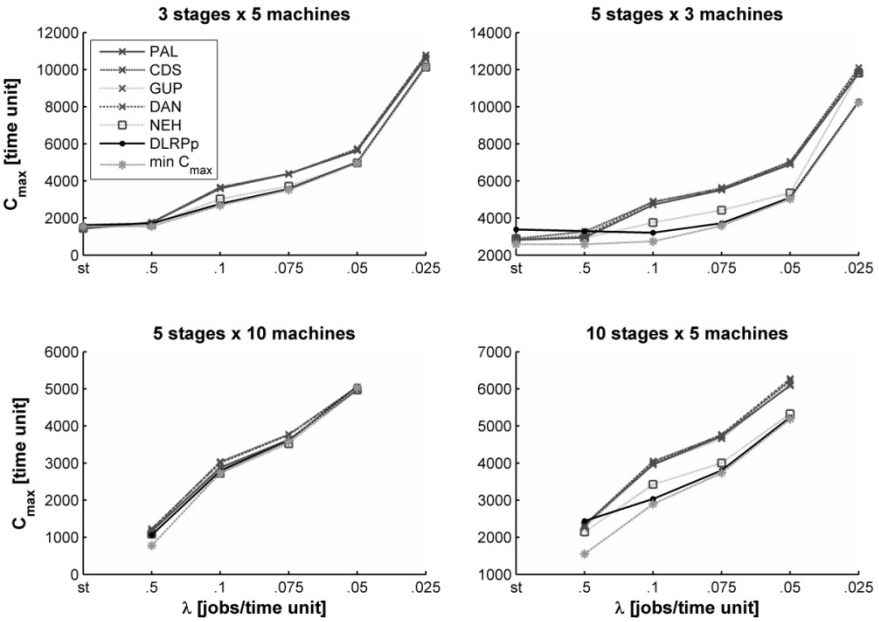

Fig. 5. Makespan against scenario size and workload.

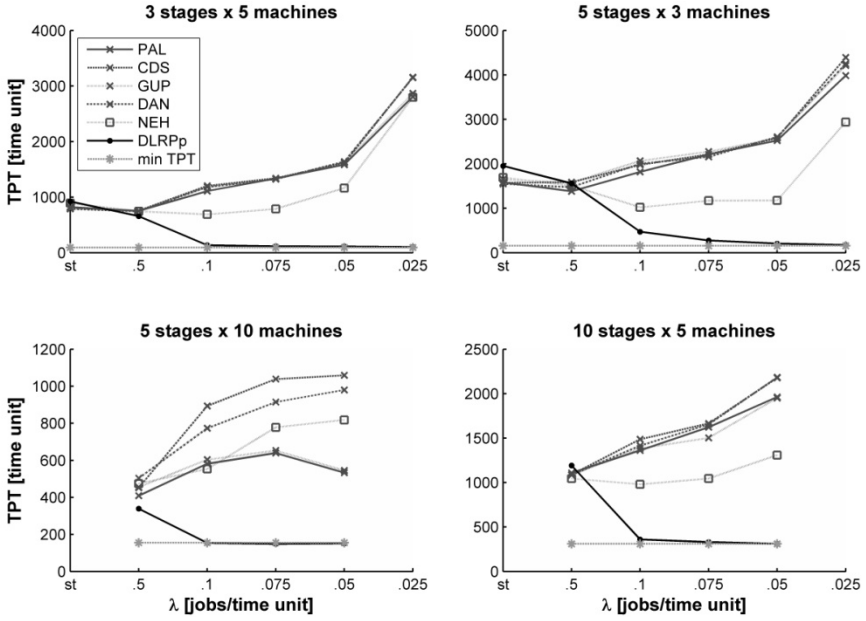

Fig. 6. Throughput time against scenario size and workload.

Rekersbrink, H.; Scholz-Reiter, B.; Zabel, C.: An Autonomous Control Concept for Production Logistics. In: Dangelmaier, W.; et al (eds.): Advanced Manufacturing and Sustainable Logistics. Springer, 2010, pp. 245-256

4 Conclusion

A new autonomous control approach for production logistics was presented and the Distributed Logistics Routing Protocol for production was described in detail. The evaluation study showed, that the DLRPp is a promising alternative to common scheduling algorithms.

For the scenarios chosen, it is shown, that the DLRPp is nearly equal the reference algorithms for small and more static scenarios. For larger and more dynamic scenarios, the DLRPp shows better results. These results are rooted in the basic difference of the concepts: On the one hand prior static and centralized planning, which leads to good results in static environments. On the other hand the ongoing and decentralized concept of autonomous control, which leads to good results in large and dynamic environments. Additionally, it is shown that the DLRPp has crucial advantages with multiple optimization goals. Additional key figures and goals like tardiness or machine utilization could easily be implemented. As an ongoing autonomous control, it is plainly able to cope with dynamic environments and process disturbances. And in contrast to scheduling algorithms, the DLRP does not need all information in advance but is able to adapt to new incoming orders and new situations.

Future research will extend these investigations to other production topologies and different DLRPp variants. The deciding points of the DLRPp mentioned above are all under current investigation. Additionally an extended evaluation study will be made, e. g. with additional reference algorithms with rolling horizon planning concepts.

The autonomous control concept DLRP is not only useful with the special scenarios presented, but it can cope with many different logistic tasks. After transportation logistics and now production logistics, it is suggestive to transfer the protocol to more tasks like assembly or warehousing. The long term vision is here to connect all concepts to one integrative autonomous control system.

Acknowledgments. This research is founded by the German Research Foundation within the Collaborative Research Centre 637 ‘Autonomous Cooperating Logistic Processes – A Paradigm Shift and Its Limitations’.

References

1. Scholz-Reiter, B. , Görges, M., Philipp, T.: Autonomously controlled production systems—Influence of autonomous control level on logistic performance. CIRP Annals - Manufacturing Technology 58-1, pp. 395-398 (2009)
2. Windt, K., Jeken, O.: Allocation Flexibility - A New Flexibility Type as an Enabler for Autonomous Control in Production Logistics. In: 42nd CIRP Conference on Manufacturing Systems (2009)
3. Scholz-Reiter, B., Teucke, M., Özsahin, M.-E., Sowade, S.: Smart Label-supported Autonomous Supply Chain Control in the Apparel Industry. In: 5th International Congress on Logistics and SCM Systems (ICLS2009), pp. 44-52 (2009)

4. Böse, F., Piotrowski, J., Scholz-Reiter, B.: Autonomously controlled Storage Management in Vehicle Logistics - Applications of RFID and Mobile Computing Systems. International Journal of RF Technologies: Research and Applications, 1-1, pp. 57-76 (2009)
5. Freitag, M., Herzog, O., Scholz-Reiter, B.: Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen. Industrie Management 20-1, 23-27 (2004)
6. Rekersbrink, H., Makuschewitz, T., Scholz-Reiter, B.: A distributed routing concept for vehicle routing problems. Logistics Research 1-1, 45-52 (2008)
7. Scholz-Reiter, B., Rekersbrink, H., Freitag, M.: Internet routing protocols as an autonomous control approach for transport networks. In: 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, pp. 341-345. (2006)
8. Philipp, T., Böse, F., Windt, K.: Evaluation of Autonomously Controlled Logistic Processes. In: 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, pp. 347-352. (2006)
9. Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., Werner, F.: An evaluation of sequencing heuristics for flexible flowshop scheduling problems with unrelated parallel machines and dual criteria. Preprint series: 05–28 (MSC: 90B35) Otto-von-Guericke-Universität Magdeburg Germany. (2005)
10. Conway, R. W., Maxwell, W. L., Miller, L. W.: Theory of scheduling. Addison-Wesley, Reading/Mass (1967)
11. Garey, M. R., Johnson, D. S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
12. Park, Y. B., Pegden, C. D., Enscore, E. E.: Survey and evaluation of static flowshop scheduling heuristics. International Journal of Production Research 22, 127-41 (1984)
13. Ruiz, R. , Maroto, C.: A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research 165, 479-494 (2005)
14. Jones, A., Rabelo, L. C.: Survey of Job Shop Scheduling Techniques. National Institute of Standards and Technology (2000)
15. Palmer, D. S.: Sequencing jobs through a multi-stage process in the minimum total time - a quick method of obtaining a near optimum. Operational Research Quarterly 16-1, 101-107 (1965)
16. Campbell, H. G, Dudek, R. A., Smith, M. L.: A heuristic algorithm for the n-job m-machine sequencing problem. Management Science 16-10, 630-637 (1970)
17. Gupta, J. N. D.: A functional heuristic algorithm for the flow-shop scheduling problem. Operations Research Quarterly 22-1, 39-47 (1971)
18. Dannenbring, D. G.: An evaluation of flow shop sequencing heuristics. Management Science. 23-11, 1174-1182 (1977)
19. Nawaz, M., Enscore, Jr. E., Ham, I.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. OMEGA 11-1, 91-95 (1983)