Scholz-Reiter, B.; Rekersbrink, H.; Wenning, B.-L.: A Distributed Routing Concept for Dynamic Flexible Flowshop Problems with Unrelated Parallel Machines. In: Proc. of 43rd CIRP International Conference on Manufacturing Systems. Wien, 2010, pp. 643-650

A Distributed Routing Concept for Dynamic Flexible Flowshop Problems with Unrelated Parallel Machines

B. Scholz-Reiter[1] (2), H. Rekersbrink[1], B.-L. Wenning[2]

[1] BIBA - Bremer Institut für Produktion und Logistik GmbH, Hochschulring 20, 28359 Bremen, Germany

[2] Communication Networks Universität Bremen, Otto-Hahn-Allee NW1, 28359 Bremen, Germany

Abstract

The flexible flowshop problem is a well known and common challenge. Extended with unrelated parallel machines and especially with the dynamic aspect of distributed release times, the problem gets very 'close to reality'. Triggered by the growing complexity of logistic systems, the paradigm of central planning is being shifted to decentralised control. As part of this, an autonomous control concept was developed for the field of transport logistics called Distributed Logistics Routing Protocol (DLRP), which has been proposed before. The work presented in this paper will focus on a new developed Routing Protocol, which was transferred from transport logistic to the dynamic flexible flowshop problem. In order to evaluate the new DLRP concept, various common scheduling algorithms are taken as reference. Problem instances are defined and solved by the DLRP and the reference algorithms. The results of this evaluation are shown and discussed.

Keywords:

Logistics, Flexible Flowshop, 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 decentralised, autonomous control strategies. 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

Scholz-Reiter, B.; Rekersbrink, H.; Wenning, B.-L.: A Distributed Routing Concept for Dynamic Flexible Flowshop Problems with Unrelated Parallel Machines. In: Proc. of 43rd CIRP International Conference on Manufacturing Systems. Wien, 2010, pp. 643-650

problems in transport logistics. This idea leads 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 decentralised, autonomous control where each logistic entity is able to interact and decide autonomously (see [1], [2], [3]).

After the suitability and performance of the DLRP concept was shown for the field of transport logistics ([2]), it was obvious to transfer the basic concept to other logistic fields, like production control. 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.

The DLRP for the production environment was developed on the basis of a dynamic flexible flowshop problem with unrelated parallel machines. The next section gives a description of this flexible flowshop problem. After that, the new routing concept DLRP for production is presented. In the fourth chapter, the scheduling algorithms which are used to evaluate the DLRP are described. The used problem instances are specified in chapter five. After that, the evaluation results are shown and discussed. The last chapter seven concludes the work and gives an outlook on future research.

2 PROBLEM DESCRIPTION

In order to evaluate the new developed distributed routing concept, an existing problem formulation from Jungwattanakit [2] is chosen. It defines a flexible flowshop problem with unrelated parallel machines and sequence-dependent setup times. In addition to this problem formulation, order types were defined here. This leads to less different production and setup times.

Consider a flexible flowshop setup with 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. Each machine has its own buffer. There are S different order types. All order types have different processing times $p^t_{m,s}$ on the different machines and they have different sequence-dependent setup times $s^t_{u,s}$. 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

Scholz-Reiter, B.; Rekersbrink, H.; Wenning, B.-L.: A Distributed Routing Concept for Dynamic Flexible Flowshop Problems with Unrelated Parallel Machines. In: Proc. of 43rd CIRP International Conference on Manufacturing Systems. Wien, 2010, pp. 643-650

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 $p^t_{m,s} = ps^t_s \big/ v^t_m$ (1)

Variables:

C_j	completion time of order j
T_j	throughput time of order j; where $T_j = C_j - r_j$

3 DISTRIBUTED LOGISTICS ROUTING PROTOCOL

The core assumption for the development of autonomous control is that every logistic entity is able to communicate with other entities and make own decisions. The basic concepts are adapted from routing algorithms that are used in wireless adhoc communication networks, where routes have to be found in dynamically changing topologies.

It is not possible to give a detailed description of the developed protocol, but on the basis figure 1, the fundamentals of the protocol can be illustrated (see [2], [3] for details). When an order enters the system, it needs a route through the production environment. It sends a route request to the next machine which fills it with necessary information and sends it ahead to all possible successional machines. These do the same until the last production step is reached. The last machine sends back all the collected information as a route reply. The order receives several route alternatives by this discovery scheme. After its route decision, the order disannounces old routes (if there was a redecision) and announces the new routes. To have a higher degree of freedom for the system, one order decides for multiple desired routes. Together with the route announcement, 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, the dispatching and the next machine for every order that leaves.

In contrast to classical scheduling algorithms, the DLRP is designed for controlling an ongoing process. It does not need all information in advance – although it would be possible to announce roughly planned future orders to the system. The structure of the DLRP leads to some curtail advantages

Scholz-Reiter, B.; Rekersbrink, H.; Wenning, B.-L.: A Distributed Routing Concept for Dynamic Flexible Flowshop Problems with Unrelated Parallel Machines. In: Proc. of 43rd CIRP International Conference on Manufacturing Systems. Wien, 2010, pp. 643-650

like possible manual interventions, estimation of future net states, implicit uncertain knowledge or an arbitrary decision process.

In the simulations studies below the decisions of the objects are made as follows. The route decision of the orders is based on the expected completion time from the route reply with some preference on already announced routes . Two routes are announced with different preferences. The decision for the next machine is based on the preference value for the different route announcements. The dispatching is based on the shortest setup time.

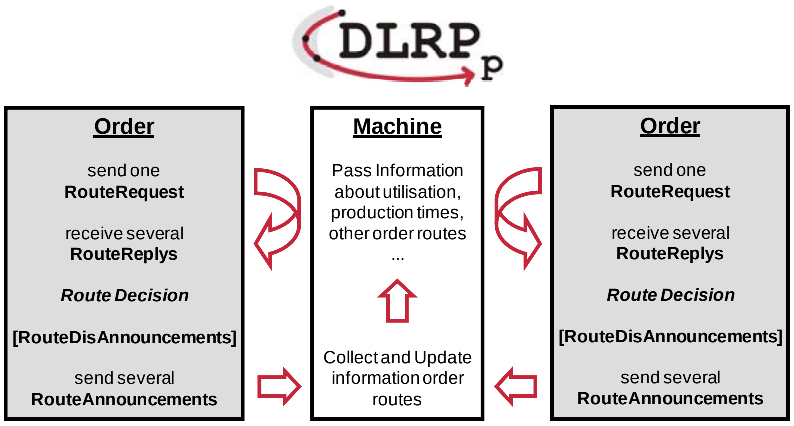

Figure 1: Basic scheme for the Distributed Logistics Routing Protocol.

4 REFERENCE ALGORITHMS

The flexible flowshop scheduling problem itself is NP-hard (see e. g. [4]). Therefore there are no algorithms known so far for finding an optimal solution for the *dynamic* flowshop scheduling problem, not to mention unrelated parallel machines.

In order to find approximate solutions, many heuristic methods were developed for the classical flexible flowshop problem. These heuristics were taken as basis for the development of heuristics for the dynamic flowshop problem. Most researchers follow the same scheme to construct a schedule for all stages and all machines. Firstly, an order sequence for the first stage is determined by an adapted *sequencing algorithm*. After that, a scheme of sorting algorithms and algorithms, which selects the best solution out of different alternatives, creates the complete schedule for all stages. For a very detailed description of this *algorithm scheme*, see [5].

Jungwattanakit, Reodecha, Chaovalitwongse and Werner made large simulation studies to evaluate this algorithm scheme with different sequencing heuristics for the problem described in 2 (see [5]). They used the following sequencing algorithms for the first stage.

- PAL: a slope index heuristic by Palmer, see [6]
- CDS: a best choice heuristic by Campbell, Dudek and Smith, see [7]

Scholz-Reiter, B.; Rekersbrink, H.; Wenning, B.-L.: A Distributed Routing Concept for Dynamic Flexible Flowshop Problems with Unrelated Parallel Machines. In: Proc. of 43rd CIRP International Conference on Manufacturing Systems. Wien, 2010, pp. 643-650

- GUP: a slope index heuristic by Gupta, see [8]
- DAN: a heuristic by Dannenbring, see [9]
- NEH: a constructive heuristic by Nawaz, Enscore and Ham, see [10]

All algorithms try to minimise the makespan ($C_{\max} = \max(C_j) - \min(r_j)$) of a given problem instance.

The mentioned algorithm scheme and the different sequencing algorithms are taken as reference algorithms to evaluate the new developed method.

5 PROBLEM INSTANCES

In order to keep the comparability, the problem instances are chosen very close to the so called ‘large size problems’ defined by Jungwattanakit, Reodecha, Chaovalitwongse and Werner in [5].

- the number of orders J can be 50, 250 or 500
- the number of order types S is set to 10, they are uniformly distributed
- the number of stages T is set to 5
- the number of parallel machines M^t is set to 3 for all t
- the standard processing times ps_s^t are integers uniformly distributed in the interval [1 50]
- the relative speeds v_m^t are uniformly distributed in [0.7 1.3]
- the setup times $s_{m,u,s}^t$ are integers uniformly distributed in [1 10]
- the release times r_j are all 0 (st, static) or a poisson process with r orders per time; r can be 0.5, 0.1, 0.075, 0.05, 0.025
- the instance name is created by J xT xM _r , e. g. 250x5x3_0.05

With three variations in the number of orders and six variations in the release time distribution, there are 18 different problem instance types. For each type, there are five different instances with varying random numbers. Table 1 shows a lineup of all 90 problem instances used for simulation.

	r					
	st	0.5	0.1	0.075	0.05	0.025
50x5x3						
250x5x3		each with 5 instances				
500x5x3						

Table 1: Problem instances used.

Scholz-Reiter, B.; Rekersbrink, H.; Wenning, B.-L.: A Distributed Routing Concept for Dynamic Flexible Flowshop Problems with Unrelated Parallel Machines. In: Proc. of 43rd CIRP International Conference on Manufacturing Systems. Wien, 2010, pp. 643-650

6 RESULTS

The 90 problem instances are solved by all five reference algorithms and by a DLRP simulation. The results of these calculations are shown in figure 2 and 3 below. Each point in the graphs represents the average of five instances. For the reference algorithms, only the best solution of all algorithms is shown. To have an estimation for a lower bound for the makespan, an average of a minimum makespan is calculated:

$$\begin{aligned} minC_{max} &= \max\begin{pmatrix} \max(r_j) + T \cdot \text{mean}(p^t_{m,s}) + T \cdot \text{mean}(s^t_{m,u,s}) \\ J \cdot \left(\text{mean}(p^t_{m,s}) + \text{mean}(s^t_{m,u,s})\right) / M^t \end{pmatrix} \\ &= \max\begin{pmatrix} \max(r_j) + 5 \cdot 25.5 + 5 \cdot 5.5 \\ J \cdot (25.5 + 5.5)/3 = J \cdot 10.33 \end{pmatrix} \end{aligned} \tag{2}$$

Similar to the makespan, an estimation for a lower bound for the throughput time is calculated:

$$\begin{aligned} minTPT &= T \cdot \text{mean}(p^t_{m,s}) + T \cdot \text{mean}(s^t_{m,u,s}) \\ &= 5 \cdot 25.5 + 5 \cdot 5.5 = 155 \end{aligned} \tag{3}$$

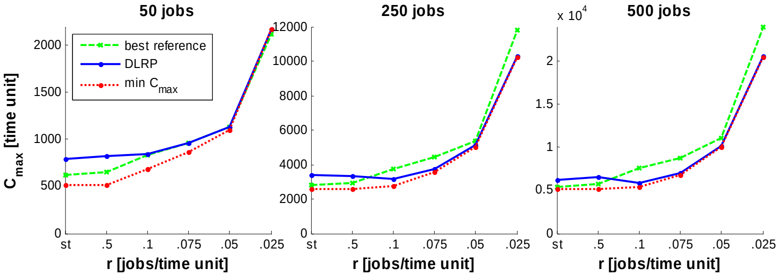

Figure 2: Makespan against scenario size and dynamics.

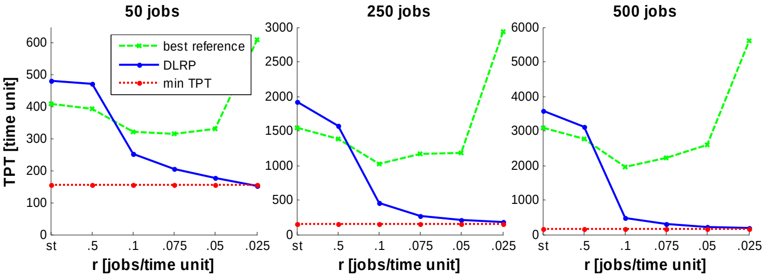

Figure 3: Throughput time against scenario size and dynamics.

Every stage is able to process three orders at the same time. With an average processing and setup time, one stage processes $3/(25.5+5.5)=0.097\approx 0.1\ \mathrm{jobs/time\ unit}$. This is a theoretical value, but it gives an idea of the capacity of the production system. In figure 3 one can see a step around this dynamics of $0.1\ \mathrm{jobs/time\ unit}$. Left to that step, the orders are arriving too fast to be processed immediately, which results in high buffer levels and throughput times. Right to the step the throughput time of the DLRP is close to the theoretical minimum, which means that the system is able to process all incoming orders without rising buffer levels. Values around $r=0.1\ \mathrm{jobs/time\ unit}$ would be a realistic system workload.

Concerning the comparison of the reference algorithms and the DLRP, one can see two tendencies in figure 2. The reference algorithms are better with more static and with smaller scenarios. As the reference algorithms are designed for scheduling of a static problem and small problem instances, this is not too surprising. On the other side, the DLRP is designed as an ongoing control system for large and dynamic problems. For larger scenarios and higher dynamics, the DLRP gets better than the reference algorithms concerning the makespan.

In figure 3 one can see the crucial disadvantage of a scheduling algorithm in a dynamic environment. All reference algorithms try to minimize the makespan without taking care about the throughput time. Therefore, the throughput time gets very high with more dynamic instances for the reference scenarios. As a control method, the DLRP regards both makespan and throughput. It is able to minimise the throughput time with growing dynamics.

7 CONCLUSIONS

For the chosen scenarios, it is shown that the DLRP is able to reach better makespan values for larger scenarios and higher dynamics than common scheduling algorithms – especially around the identified realistic workload for the system. This lies in the basic difference of both concepts: static and centralised planning on the one side and autonomous control on the other side. Additionally, the DLRP is able to meet multiple goals at once. This is shown by the good results for makespan and throughput time. Classical scheduling algorithms are not able to cope with multiple goals.

For realistic production planning situations, the new routing concept has several advantages. In contrast to scheduling algorithms, the DLRP is able to implement and fulfil multiple goals like makespan, throughput time, tardiness etc. As an ongoing autonomous control, it is plainly able to cope with dynamic environments and process disturbances. The computational effort for large scenarios is short for the DLRP compared to scheduling. The scheduling algorithms need all information in advance, the DLRP does not need this and is able to adapt to new situations.

Future research will extend these investigations to other production topologies and different DLRP variants. It is also reasonable to evaluate the DLRP against scheduling algorithms with rolling planning horizons.

8 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’.

9 REFERENCES

[1] Freitag, M., Herzog, O., Scholz-Reiter, B., 2004, Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen, Industrie Management, 20(2004)1: 23-27

[2] Rekersbrink, H., Makuschewitz, T., Scholz-Reiter, B., 2008, A distributed routing concept for vehicle routing problems, Logistics Research, 1/2008: 45-52

[3] Scholz-Reiter, B., Rekersbrink, H., Freitag, M., 2006, Internet routing protocols as an autonomous control approach for transport networks, 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, Proceedings: 341-345

[4] Garey, M. R., Johnson, D. S., 1979, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA

[5] Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., Werner, F., 2005, An evaluation of sequencing heuristics for flexible flowshop scheduling problems with unrelated parallel machines and dual criteria, Germany: Otto-von-Guericke-Universität Magdeburg, Preprint series: 05–28 (MSC: 90B35): 1-23

[6] Palmer, D. S., 1965, 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

[7] Campbell, H. G, Dudek, R. A., and Smith, M. L., 1970, A heuristic algorithm for the n-job m-machine sequencing problem, Management Science,16(10): 630-637

[8] Gupta, J. N. D., 1971, A functional heuristic algorithm for the flow-shop scheduling problem, Operations Research Quarterly, 22(1): 39-47

[9] Dannenbring, D. G., 1977, An evaluation of flow shop sequencing heuristics, Management Science, 23(11):1174-1182

[10] Nawaz, M., Enscore, Jr. E., Ham, I., 1983, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, OMEGA 11(1): 91-95