Chapter 11
Potentials and Limitations of Autonomously Controlled Production Systems

Bernd Scholz-Reiter, Michael Görges, and Henning Rekersbrink

11.1 Introduction

Manufacturing enterprises are increasingly challenged by a dynamic and volatile business environment. Driving forces in this context are for example an increasing demand of customers for individualized goods, short delivery times and a strict adherence to due dates. Moreover, internal factors like machine breakdowns or rush orders lead to additional dynamics. In order to sustain competitive, manufacturing enterprises have to react promptly to these changes. Conventional centralized production planning and control methods are not able to cope with these dynamics in an appropriate manner [8]. In this context the application of novel decentralized approaches, like autonomous control, seems to be promising. The concept of autonomous control aims at shifting decision making capabilities form the total system to its elements [24]. This approach enables autonomous decision making of intelligent logistic objects. The term intelligent logistic object is broadly defined and covers material objects (e.g., parts in a shop floor) as well as immaterial object (e.g., production orders). On this basis the concept of autonomous control aims at increasing robustness and performance of logistic systems [25].

Previous studies have shown the effectiveness of autonomous controlled production systems in highly dynamic situations compared to conventional methods. Nevertheless, conventional planning methods tend to outperform autonomous control in well-defined situations with less dynamics [15]. This paper addresses the boundaries and the potentials of autonomous control in different static and dynamic situations. The main hypothesis in this context reads as follows: Autonomous control performs best in complex and dynamic situations, while conventional planning methods outperform autonomous control under less dynamic and static conditions.

B. Scholz-Reiter, M. Görges (✉), and H. Rekersbrink
Department of Planning and Control of Production Systems, BIBA, University of Bremen, Bremen, Germany
e-mail: bsr@biba.uni-bremen.de

M. Hülsmann et al. (eds.), *Autonomous Cooperation and Control in Logistics*,
DOI 10.1007/978-3-642-19469-6_11,

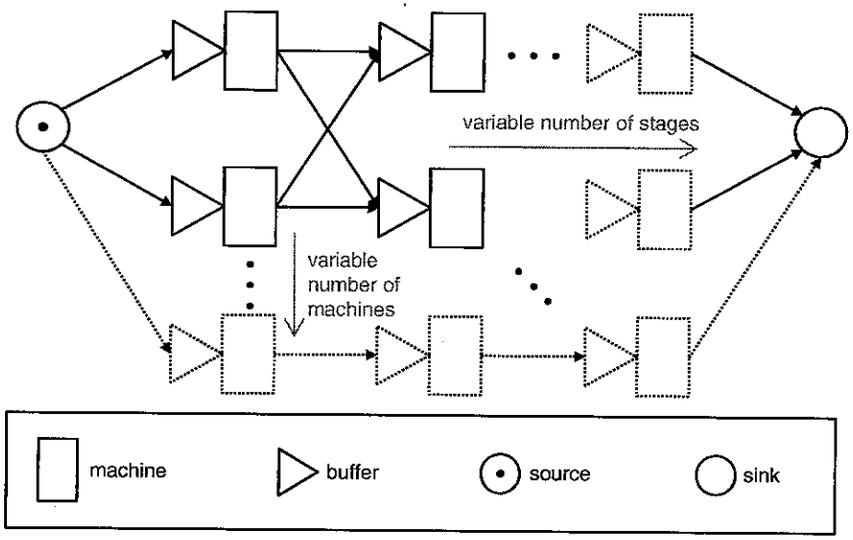

Fig. 11.1 Scheme of Flexible Flow Shop (FFS) problems [10]

Therefore, this contribution evaluates two different autonomous control methods and twelve conventional scheduling heuristics in a Flexible Flow Shop (FFS) environment. The FFS scheduling problem is a well known and common task in operations research. Basically, the FFS structure which is depicted in Fig. 11.1 comprises several production stages with a certain amount of parallel machines per stage [1].

Jobs running through the FFS system have to pass each production stage once. In addition to the basic version of the FFS problem, practice-oriented extensions with unrelated parallel machines and sequence dependant setup times are considered. The analysis covers scenarios with different degrees of structural complexity and dynamics. In this respect the structural complexity refers to the number of production stages and the number of parallel machines per stage. Moreover, different degrees of dynamics are modeled by varying inter-arrival times of the incoming work load: The dynamic situations are characterized by distributed inter-arrival times of the jobs. By contrast, in the static situation all jobs are released at once. The structure of this contribution is as follows: Sect. 11.2 briefly describes the FFS problem class and its extensions. On this basis Sect. 11.3 presents classical scheduling heuristics for this problem class. Subsequently, Sect. 11.4 provides information about the concept of autonomous control in manufacturing and presents particular autonomous control methods, which are used in this evaluation study. The results of the evaluation study are presented and discussed in Sect. 11.5. Finally, Sect. 11.6 summarizes the potentials and the limitations of autonomous control in manufacturing and gives an outlook with further research fields.

11.2 Flexible Flow Shop Problems

Within the operation research domain, the FFS scheduling problem is a common and well-known task. Generally, this problem class aims at sequencing and assigning a set of jobs to a set of production resources in an optimal manner [1]. However, for most instances of this problem class optimal solutions cannot be found in an applicable computational time. The main reason for this is the complexity of the problem class, which is NP-hard [3]. Hence, heuristic approaches were developed and applied to these problems in the past in order to derive acceptable solutions. This section gives a detailed description of the structure of this problem class. Subsequently, different solution heuristics are introduced.

The FFS problem is characterized by a set of production stages, which are sequentially arranged. Each of these production stages comprises at least one machine or different parallel machines. Generally, a FFS scenario has at least one production stage with more than one machine. Figure 11.1 depicts an example of a FFS scenario. Jobs running through this network have to pass each production stage once. Each job can be processed by every machine on a production stage. A practice-oriented extension of this basic scenario is the extension of unrelated parallel machines and setup times [5]. The term unrelated parallel machines indicates, that the processing times and the setup times on a production stage may vary between the machines for a particular job type. With regard to scheduling problems in practice, this formulation is considered to be realistic. Jungwattanakit et al. (2008) present a problem formulation of the FFS problem, which is adapted for this contribution as follows [6]:

Parameters and Variables:

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

Logistic target measures

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

According to this formulation, there are T stages with M^t unrelated machines at each stage t. Moreover, there are S different types of jobs, which have all different

processing times $p^t_{m,s}$ and setup times $s^t_{m,u,s}$ on the machines. Ever job has a predefined release time, denoted by r_j.

11.3 Scheduling Heuristics

The FFS problem in its basic form is NP-hard [9]. Accordingly, optimal solutions can only be determined for very small and less complex scenarios in an applicable computational time. Many different heuristic approaches were developed in the past, to generate suitable solutions for the FFS problem. As far as the FFS with the extension of unrelated parallel machines and setup times is concerned, more sophisticated heuristics are necessary. Jungwattanakit et al. (2005) introduces a heuristic approach containing a set of constructive algorithms and a greedy algorithm [5]. All constructive algorithms are based on heuristic for the flow shop scheduling. They construct an initial sequence for the first production stage. Jungwattanakit et al. (2005) propose the following sequencing algorithms for the first stage: PAL (a slope index heuristic by Palmer), CDS (a best choice heuristic by Campbell, Dudek and Smith), GUP (a slope index heuristic by Gupta), DAN (a heuristic by Dannenbring), NEH (a constructive heuristic by Nawaz, Enscore and Ham) [5].

Subsequently, a greedy algorithm allocates the jobs to the machines at the stages. It assigns jobs to machines considering the setup state of a machine, the processing time of a job on the machine and the idle time of the machines. This algorithm repeats for every stage of the scenario, until a complete schedule is generated for all stages. For a detailed description of the constructive and the greedy algorithm can be found in Jungwattanakit et al. (2005) and Jungwattanakit et al. (2008) [5, 6].

In order to improve the results of this basic procedure with construction of an initial sequence, which is followed by an incremental assignment to machines, Jungwattanakit et al. (2009) propose a coupling of this basic procedure with a genetic algorithm (GAL) [7]. This algorithm takes the sequence for the first stage as genomes. It creates an initial population out of the results of all basic heuristics. The optimization results of the genetic algorithm outperform the solutions of the basic procedure. Thus, the GAL is taken as reference benchmark of all scheduling heuristics in this evaluation study.

Due to the expected shortcomings of these scheduling heuristics in dynamic situations, Scholz-Reiter et al. (2010) developed adapted versions of these heuristics for rolling horizons (rPAL, rCDS, rGUP, rDAN, rNEH and rGAL) [18]. The rolling horizon scheduling heuristics (rSCD) divide the entire planning horizon chronologically into sub-planning-horizons, which comprise 25 jobs each. These sub-instances are solved sequentially. The final schedule is generated by merging the sub-schedules.

Additionally, a further benchmark is considered. It is the lower bound of the evaluation. This boundary contains the theoretical minimum values, which can be archived in a scenario. For $C_{\max}$ minimum is determined by:

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

Similar to $C_{\max}$ the minimum of the TPT is estimated:

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

In contrast to the $C_{\max}$ and the TPT, the best possible utilization of a scenario is 100%. The lower bound is used for estimating the quality of the simulation results. Note, that these general lower bounds are theoretical values. It is assumed, that there is no feasible solution for realizing these lower bounds in the particular scenarios.

11.4 Autonomous Control in Manufacturing

The Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes – a Paradigm Shift and its Limitations" gives the following comprehensive definition of autonomous control: "Autonomous control describes processes of decentralized decision-making in heterarchical structures. It presumes interacting elements in non-deterministic systems, which possess the capability and possibility to render decisions independently. The objective of autonomous control is the achievement of increased robustness and positive emergence of the total system due to distributed and flexible coping with dynamics and complexity." [24]. According to this definition, the concept of autonomous control is characterized by a shift of decision making capabilities to the logistic objects. Generally, this concept can be applied to various logistic areas, like the intelligent container [4], transport planning [11, 12, 19], manufacturing systems [13] or entire production networks [16].

In the context of manufacturing systems the implementation of autonomous control aims at enabling jobs to find routes through a production system referring to their own logistic targets. This kind of autonomous decision making and the corresponding interactions between the logistic objects aims at a generating a self-organizing behavior which increases the robustness and the performance of the system. This self-organisation is a called emergent behaviour of a complex dynamic system and not derivable from single characteristic [21–23].

Different autonomous control methods were developed in the past. These methods have shown promising results concerning the system's ability to cope with dynamics and unforeseen disturbances e.g. [13]. Scholz-Reiter et al. (2009) compared different autonomous control methods with a conventional production planning and control (PPC) approach in a real data based model of a production system [15]. It was shown that different autonomous control methods outperform the PPC approach in a highly dynamic situation. Moreover, this study confirmed that different autonomous methods lead to variations in the systems performance, depending on logistic target prioritization.

Scholz-Reiter et al. (2010) propose a classification of different autonomous control methods according to their information horizon [18]. It refers to the approach of collecting and processing necessary information for decision making of autonomous control methods. This differentiation identifies local information methods and information discovery methods. All methods of both groups enable local, autonomous decisions of jobs, but they use different information horizons for the decision making process. Local information methods collect information exclusively from the direct neighborhood (buffers or machines). By contrast, information discovery methods collect selected information from the entire production system. Usually, these methods do not discover all available data. It is rather directed to all relevant information.

11.4.1 Local Information Methods

Local information methods avoid elaborate discovery procedures. Generally, they aim at rendering suitable local decision with less computational effort. All local information methods have in common that they focus only on information, which are available from the direct local environment (e.g., data form succeeding buffers and machines). The underlying decision procedures are rather simple, compared to information discovery methods. According to a classification, introduced by Windt and Becker (2009), these local information methods can be grouped as follows: rational strategies, bounded rational strategies, combined strategies [26]. This classification is based on the underlying decision mechanisms used by the different autonomous control methods. Rational strategies utilize rational measures for decision making. This means for example estimated buffer and waiting times or due dates of orders.

The queue length estimator method (QLE) is a rational strategy. Jobs using the QLE method are able to collect information about the states of all direct succeeding alternative production resources. These jobs estimate the respective processing and waiting times. For further processing, the respective job will choose the alternative with the lowest estimated waiting and processing time [13]. By contrast, biologically inspired methods belong to the bounded rational strategies. They aim at transferring fundamental mechanisms of biologic self-organizing systems to autonomous decision making methods. There are methods, which use the foraging behavior of ants and honey bees or the chemotaxis movement principles of bacteria [2, 14, 17]. In order to keep the evaluation study comprehensible, the QLE method is implemented to the scenarios of the FFS as a representative of the local information methods.

11.4.2 Information Discovery Methods

In contrast to local information methods, information discovery methods focus on globally distributed information, which is available in the production system. They

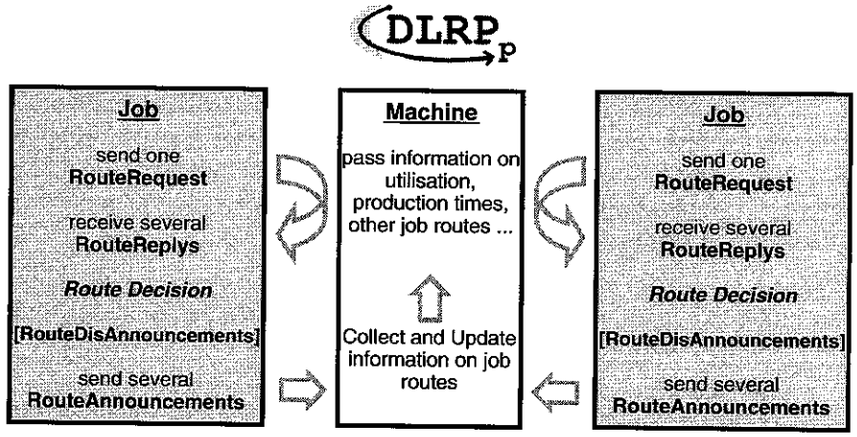

Fig. 11.2 Schematic interactions between jobs and machines [12]

conduct higher computational efforts to collect data from different network stages. The Distributed Logistic Routing Protocol (DLRP) belongs to the class of information discovery methods. In its origins, the $DLRP_t$ was designed for transport logistics [12]. The index t indicates its design for transport logistic purposes. This protocol is inspired by communication algorithms in wireless ad hoc networks. In the transport logistic context, the $DLRP_t$ enables autonomous routing of goods and the corresponding vehicles. However, the application of these principles is not limited to autonomous control of transport logistic processes. Rekersbrink et al. (2010) transferred the approach of the DLRP to autonomous control of shop floors environments [12].

The $DLRP_p$ for production systems has two different object types: machines and jobs. Figure 11.2 illustrates the interactions between both object types within the $DLRP_p$.

In order to find a route through a production system, a job sends a route request to all succeeding machines. The machines fill all necessary information, like waiting times, setup states or urgencies, into the route request. Subsequently, they pass the request to all further successors. This process repeats until the last production stage is reached. The last machine sends back the information as route reply to the job. On this basis, the job is able to receive the route replies and to select a route according to its individual target preferences. After deciding for one or more routes the job sends route announcements to the machines involved. In contrast to classical scheduling methods, the $DLRP_p$ supports a real-time autonomous decision making process, which goes beyond pure pre-planning procedures. Hence, jobs are able to revise previous decisions in terms of generating route disannouncements and new route announcements. Furthermore, the second object class (machines) is able to select jobs form the buffer according to their target preferences.

Due to the complexity of the $DLRP_p$, this contribution gives only a brief description of the protocol. For a detailed description of the $DLRP_p$ for production systems see [12].

In this contribution the $DLRP_p$ is modeled as a representative of the information discovery methods. Jobs using the $DLRP_p$ will base their routing decision on the expected completion time. Each job sends two route announcements with different preference values. Additionally, the machines select jobs from the corresponding buffer according to the shortest setup time rule.

11.5 Evaluation and Results

Different instances of the FFS scheduling problem are used to compare the performance of the classical scheduling heuristics, the local information method and the information discovery method concerning the structural complexity and different degrees of dynamics. The following presents the concrete parameterization of the instances used. Subsequently, the simulation results are presented and discussed.

11.5.1 Problem Instances

The problem instances used are based on the "large size instances" introduced by [5]. There are $S = 10$ different job types. Every instance contains $J = 250$ jobs, in total. A uniform distribution is used for assigning jobs to job types. A uniform distribution is used as well, to obtain the standard processing times, the relative speed of the machines and the setup times. Table 11.1 summarizes the system related time parameters and the corresponding interval.

The structural configuration of the scenarios are varied. There are instances with $T = 3, 5$ and 10 production stages. Moreover, the number of parallel machines per stage is varied. Accordingly, there are instances with $M^t = 3, 5$ and 10 machines per stage.

In order to model different degrees of dynamics Sung and Kim (2002) used a uniform distribution with varying interval ranges [20]. Similar to this approach, the

Table 11.1 Parameterization of instances

Parameter	Description	Distribution	Interval/values
$s^t_{m,u,s}$	Fixed	Uniform distribution	[1 50]
ps^t_s	Fixed	Uniform distribution	[1 10]
v^t_m	Fixed	Uniform distribution	[0.7 1.3]
T	Variable for scenarios	Const.	{3, 5, 10}
M^t	Variable for scenarios	Const.	{3, 5, 10}
λ	Variable for scenarios	Exponential distribution	{st, 0.5, 0.1, 0.075, 0.05}

inter-arrival times of jobs $r_{j+1} - r_j$ are set to an exponential distribution with a mean value λ. Consequently, the arrival process is a poisson process. In order to model different degrees of dynamics, λ is varied form $\lambda = st, 0.5, 0.1, 0.075$ to 0.05. Lambda denotes dynamic aspects of the incoming work load. In the static situation ($\lambda = st$, this means $\lambda \approx \infty$) all jobs have the release time $r_j = 0$, which means that all jobs enter the system simultaneously. By contrast, a decrease of λ leads to more extended arrival interval. Thus, the workload of the system depends on λ.

11.5.2 Evaluation

Figure 11.3 presents the results concerning C_{max} of both autonomous control methods and for the genetic algorithm (GAL). Each graph of Fig. 11.3 shows the results of C_{max} in a particular scenario configuration against the different degrees of

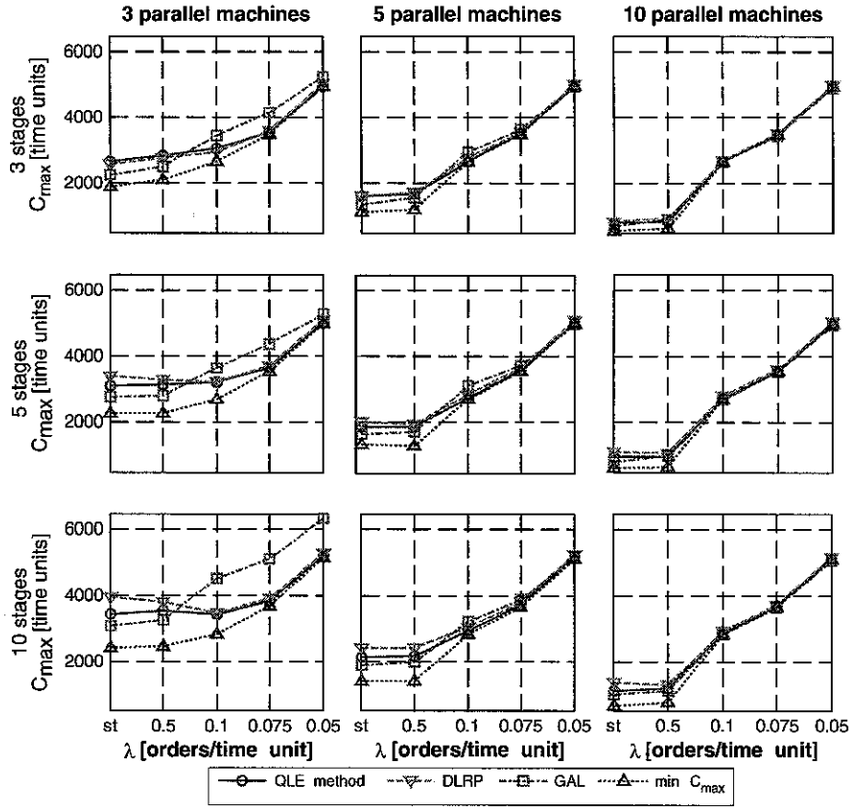

Fig. 11.3 Simulation results for makespan

dynamics. In addition, each graph contains the theoretical minimal values for the respective scenario.

According to Fig. 11.3, the performance of the scheduling method (GAL) and both autonomous control methods depends on the degree of dynamics. In all cases, the GAL outperforms the QLE method and the DLRP in the static ($\lambda = st$) and the nearly static situation ($\lambda = 0.5$). Here, the GAL performs best. However, with increasing λ the performance of both autonomous control methods gets closer to the performance of the GAL. Between $\lambda = 0.5$ and $\lambda = 0.1$ there is an intersection of the autonomous control result curves and the GAL curve. When comparing the simulation results with the theoretical minimal values another effect can be observed. The gap between the minimum and the obtained results becomes smaller with an increasing degree of dynamics. For example, in the scenario with five stages and three parallel machines the gap is 21.46% for the GAL (36.82% for the QLE and 48.79% for the DLRP) for $\lambda = st$. By contrast, the gap is for $\lambda = 0.075$ for the GAL 0.55% (0.08% for the QLE and 0.2% for the DLRP). This can be explained by the incoming workload, which differs according to λ. For lower values of λ the mean of inter-arrival times get bigger.

Hence, the system is under-utilized in this area. Figure 11.3 shows that this effect occurs in every scenario. This effect is more dominant in scenarios, which offer more parallel machines (last column of graph in Fig. 11.3). Due to this utilization effect, the performance of the autonomous control methods and the scheduling methods gets similar. Nevertheless, the explained connection between the methods performance and the dynamics persists, but with smaller differences. These effects are not limited to the results of $C_{\max}$. The systems utilization shows similar results. Figure 11.4 depicts exemplarily the utilization of the scenario with ten stages and three parallel machines per stage by showing the composition of the corresponding $C_{\max}$ values. It differentiates between realized processing times, retooling times and idle times of the machines in the respective simulation runs. The sum of these values is the makespan, which is consequently the height of the bars in Fig. 11.4. Accordingly, Fig. 11.4 give information about the systems utilization, which is the relationship between realized processing times and makespan. Figure 11.4 confirms the observed under-utilized system behavior for small values of λ. There is a step in the idle times form $\lambda = 0.075$ and $\lambda = 0.05$. This indicates that the system's capacity is not fully utilized.

Furthermore, Fig. 11.4 provides information about the methods ability to minimize processing times, retooling times and idle times. The realized processing times of all methods are very similar for all degrees of λ, but their performance differs in respect of idle times and retooling times. Especially in the static situation the GAL outperforms both autonomous control methods. It is able to construct schedules with less retooling and idle times compared to the QLE and the DLRP. However, the DLRP leads to the lowest retooling times in the static situation at an expense of longer idle times. The highest degree of retooling times can be found for the QLE method. The same effects can be observed in the nearly static situation $\lambda = 0.5$. Form this point on an increase of λ leads to rising idle times of the scheduling heuristic. The GAL is not able to assign the incoming workload to the machines in

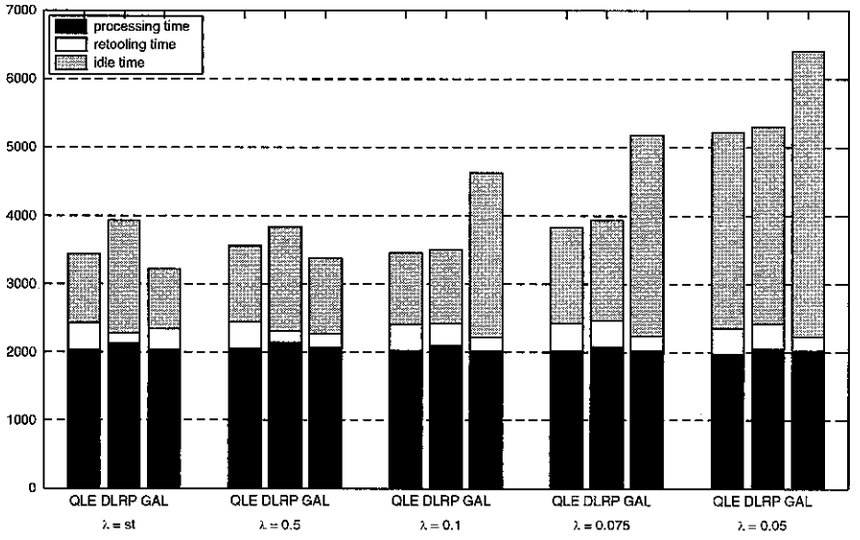

Fig. 11.4 Composition of makespan results of all methods for varying λ in the scenario with ten stages and three parallel machines

an appropriate manner. It focuses on minimizing the setup times. Thus, the GAL assigns jobs to buffers and machines with jobs of the same type.

Consequently, this leads to longer queues, waiting times of jobs and idle times of machines. By contrast, both autonomous control methods perform better in the dynamic situation. Despite longer retooling phases, they are able to harmonize the flow of jobs.

Figure 11.5 confirms this. It presents the WIP over time of different simulation runs for the QLE method, the DLRP and the GAL in an exemplary scenario with five stages and ten parallel machines for varying values of $\lambda(\lambda = 0.5, \lambda = 0.1$ and $\lambda = 0.075)$. In the nearly static situation ($\lambda = 0.5$) all methods collect a high level of WIP up to a certain maximum. At this point, all jobs are assigned to the machines and no new jobs arrive. Accordingly, the WIP level is processed constantly and the WIP decrease with time. In the nearly static situation the incoming work load is bigger than the system's capacity. This explains results with growing WIP levels. Nevertheless, the GAL performs best in this situation. The GAL decreases the WIP faster than both autonomous control methods. This leads to the lowest $C_{\max}$ compared to the autonomous control methods.

In more dynamic situations ($\lambda = 0.1$ and $\lambda = 0.075$) the results are different: The GAL still builds up a higher degree of WIP, but the QLE method and the DLRP do not. Both autonomous control methods are able to distribute the incoming work load more evenly to the available machines. Nevertheless, the GAL leads to the shortest $C_{\max}$ for $\lambda = 0.1$. With regard to the realized WIP levels, both autonomous control methods perform better in this situation. This effect can be observed more

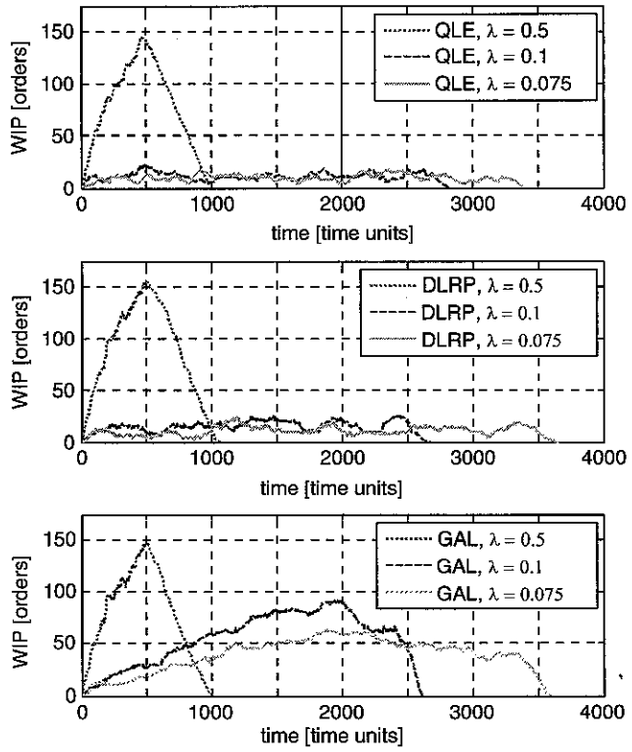

Fig. 11.5 WIP over time for the QLE method, the DLRP and the GAL

clearly for $\lambda = 0.075$. Here, the WIP levels and C_{max} are lower than those of the GAL. In this particular scenario, the QLE leads to the shortest C_{max} for $\lambda = 0.075$. But in general both autonomous control methods perform quite similar. The difference of C_{max} between the DLRP and the QLE method is 6.01% for $\lambda = 0.075$. These results show that the classical scheduling heuristics are appropriate in static situation, which tend to be over-utilized. While autonomous control methods are applicable in more dynamic situations.

Figure 11.6 strengthens these findings. It presents the average job related throughput times (TPT) for all scenarios and all methods. For reasons of comparability Fig. 11.6 contains additionally information about the lowest possible TPT. Concerning the TPT, the GAL has naturally some shortcomings. Due to the underlying heuristics, the GAL does not focus on optimizing the TPT. By contrast, the introduction of rolling planning horizons helps to overcome these shortcomings. Thus, Fig. 11.6 presents additionally the results of the rolling horizon genetic algorithm (rGAL), in order to give a comprehensible comparison.

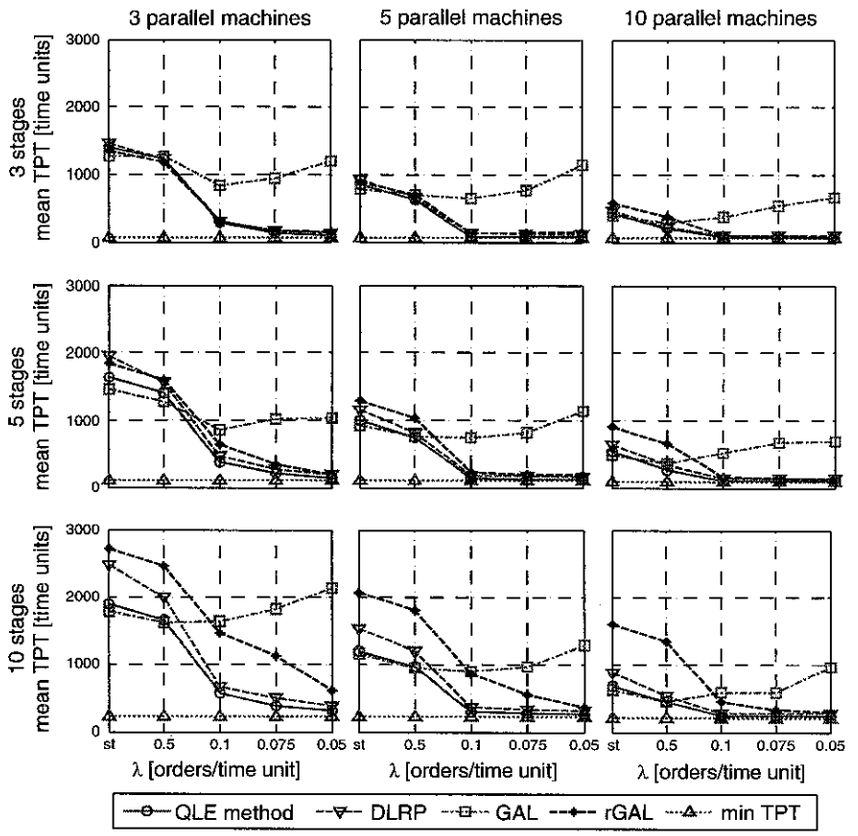

Fig. 11.6 TPT results of all scenarios

Figure 11.6 confirms the finding stated before: in the static and nearly static situation the GAL performs best concerning the achieved TPT. Similar to the results of C_{max}, the gap between both autonomous control methods and the GAL is narrower for scenarios with less production stages ($T = 3$) compared to scenarios with more production stages ($T = 10$).

There is a huge deviation between the results of the GAL, QLE method, the DLRP and the theoretical minimum values in the static situation. Again, this confirms that the system is over-utilized in this area. This changes with increasing dynamics. In the most scenarios the TPT of the GAL remains at the same level or even increases slightly with λ. This can be explained by the optimization objectives of the GAL described above. Its primary target is to optimize the makespan. The TPT is a subordinate objective for the GAL. The rGAL performs better than the GAL concerning the TPT in this dynamic situation. Due to the segmentation of the planning horizon into sub horizons, the rGAL optimizes the makespan of

the sub-horizons, which leads to a lower TPT in the total schedule. By contrast the rGAL performs even worse in the static situation. Especially in scenarios with 10 stages this effect can be clearly seen. The lowest deviation between obtained results and theoretical minimum in the static situation achieves the GAL.

The rGAL performs better with an increase of λ: A step can be observed in the TPT achieved by the rGAL between $\lambda = 0.5$ and $\lambda = 0.075$. A similar effect is obtained for both autonomous control methods. This indicates, that the utilization state of the system changes in this area from over-utilized to under-utilized situation. Accordingly, there are shorter queues at the machine buffers, which leads to a lower TPT and a more harmonic material flow. Comparing both autonomous control methods and the rGAL in the static situation one can notice that in all cases at least one of the autonomous control methods outperforms the rGAL. Especially, in the scenarios with a higher degree of structural complexity (for example $T = 10$ and $M^t = 10$) the rGAL performs worse. Due to the segmentation and the partial construction of the schedules, the rGAL performs worse for a lower degree of utilization. In this dynamic situation both autonomous control methods perform best. Furthermore, structural scenario configuration has a curial impact on the performance of the rGAL. With an increasing number of production stages the gap between the theoretical minimum values and the TPT of the rGAL grows. This structural impact is less in the case of autonomous control. In the dynamic situation the TPT of both autonomous control methods remains on a constant level with a smaller gap to the theoretical minimum value.

11.5.3 Conclusions

The results concerning C_{max}, processing times, retooling times, WIP and TPT show that for the investigated scenarios different scheduling and control methods are suitable. In very static situations with an incoming workload, which is above the system's capacity, conventional scheduling methods perform best. The GAL provides superior results concerning C_{max} and TPT in these situations, independent of the scenarios configuration (number of stages or number of parallel machines). However, with an increase of dynamics both autonomous control methods perform better regarding C_{max} and the TPT. Especially, the results of the TPT reveal shortcomings of the GAL to schedule an appropriate flow of jobs through the system. Furthermore, its TPT performance decreases with increasing dynamics in some cases. The extension with rolling horizons (rGAL) attenuates these shortcomings in the TPT performance. Despite, both autonomous control methods outperform the rGAL in dynamic situations. In a more general context, these results confirm the hypothesis stated in the outline: The scheduling algorithms perform best in a static situation, where all information are available. In this static context autonomous control reveals short comings. Due to the idea of autonomous control, which is characterized by interactions and autonomous decision making of intelligent logistic objects in dynamic production environments, this concept is not suitable to

static situations. According to the results of this contribution the main application potential of autonomous control is in complex and dynamic systems.

A comparison of the autonomous control methods shows a similar performance of both methods. The QLE method performs slightly better than the DLRP concerning the makespan and the TPT, while the DLRP leads to shorter retooling sequences. Due to its ability to discover very complex interconnected production systems it can be expected, that the DLRP performs better in more complex scenarios. This refers to scenarios like sparse production networks. In such networks routing decisions cannot be revised at any stage. Due to the information discovery and the corresponding forecast of succeeding states, the DLRP seems to be favorable in these cases.

11.6 Summary and Outlook

This contribution investigated the potentials and the limitations of autonomous cooperating logistic processes in production environments. Therefore, the FFS problem formulation was chosen and different centralized scheduling algorithms for this problem class were introduced. Besides these scheduling algorithms, the evaluation includes two different autonomous control methods. Both methods differ in their information acquirement and information processing procedures. The QLE methods focus solely on local information, while the DLRP is able to discover information distributed in the shop-floor environment. In order to evaluate the scheduling heuristics and the autonomous control methods in respect of their performance in different dynamic situations a set of simulation experiments was defined. Varying degrees of dynamics were modeled by the arrival process of the incoming work load. With regard to variations in the dynamics, the hypothesis stated in the outset was confirmed: the application of autonomous control is limited to dynamic scenarios. In static scenarios classical scheduling heuristics perform best. In situations with an incoming work load, which is higher than the system capacity, classical algorithms are superior to autonomous control methods. By contrast, autonomous control deploys its potential in dynamic situations. Compared to the scheduling algorithms both autonomous control methods allow a continuously flow of jobs through the system, while the scheduling algorithms build up high WIP levels over time.

These results encourage for further research in this area. In the next steps of the evaluation of autonomous control extensions like machine breakdowns or rush orders will be addressed. In particular, the effects of this extension on the performance of autonomous control and scheduling methods will be investigated in order to identify further limitations and potentials of autonomous control in production environments. Another research area is the evaluation of autonomous control in combined production and transport scenarios.

References

1. Allahverdi A, Ng CT, Cheng TCE, Kovalyov M (2008) A survey of scheduling problems with setup times or costs. Eur J Oper Res 187(3):985–1032
2. Armbruster D, de Beer C, Freitag M, Jagalski T, Ringhofer C (2006) Autonomous control of production networks using a pheromone approach. Phys A 363(1):104–114
3. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco
4. Jedermann R, Moehrke A, Lang W (2010) Supervision of banana transport by the intelligent container. In: Kreyenschmidt J (ed) Coolchain-management. 4th International Workshop. University Bonn, Bonn, 2010, pp 75–84
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. Preprint series: 05–28 (MSC: 90B35) Otto-von-Guericke-Universität Magdeburg Germany
6. Jungwattanakit J, Reodecha M, Chaovalitwongse P, Werner F (2008) Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Int J Adv Manuf Technol 37(3):354–370
7. Jungwattanakit J, Reodecha M, Chaovalitwongse P, Werner F (2009) A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Comput Oper Res 36(2):358–378
8. Kim JH, Duffie NA (2004) Backlog control for a closed loop PPC system. Ann CIRP 53(1):357–360
9. Pinedo ML (2008). Scheduling – theory, algorithms, and systems. Springer, New York
10. Quadt D, Kuhn H (2007) A taxonomy of flexible flow line scheduling procedures. Eur J Oper Res 178(3):686–698
11. Rekersbrink H, Makuschewitz T, Scholz-Reiter B (2009) A distributed routing concept for vehicle routing problems. Logist Res 1(1):45–52
12. Rekersbrink H, Scholz-Reiter B, Zabel C (2010) An autonomous control concept for production logistics. In: Dangelmaier W et al (eds) Advanced manufacturing and sustainable logistics, pp 245–256. Springer, Berlin
13. Scholz-Reiter B, Freitag, M, de Beer C, Jagalski T (2005) Modelling and analysis of autonomous shop floor control. Proceedings of 38th CIRP International Seminar on Manufacturing Systems. Florianopolis, Brazil
14. Scholz-Reiter B, Jagalski T, Bendul J (2007) Autonomous control of a shop floor based on bee's foraging behaviour. In: Haasis HD, Kreowski HJ, Scholz-Reiter B. (eds) First International Conference on Dynamics in logistics. LDIC 2007. Springer, Berlin, Heidelberg, pp 415–423
15. Scholz-Reiter B, Görges M, Philipp T (2009) Autonomously controlled production systems – influence of autonomous control level on logistic performance. CIRP Ann Manuf Technol 58(1):395–398
16. Scholz-Reiter B, Dashkovskiy S, Görges M, Naujok L (2010) Stability analysis of autonomously controlled production networks. Int J Prod Res. http://www.informaworld.com/10.1080/00207543.2010.505215 (accessed 07 December 2010)
17. Scholz-Reiter B, Görges M, Jagalski T, Naujok L (2010) Modelling and analysis of an autonomous control method based on bacterial chemotaxis. 43rd CIRP International Conference on Manufacturing Systems 2010 (ICMS 2010). Neuer Wissenschaftlicher, Wien, pp 699–706
18. Scholz-Reiter B, Rekersbrink H, Görges M (2010) Dynamic flexible flow shop problems – scheduling heuristics vs. autonomous control. CIRP Ann Manuf Technol 59(1):465–468
19. Schönberger J, Kopfer H (2009) Online decision making and automatic decision model adaptation. Comput Oper Res 36(6):1740–1750
20. Sung CS, Kim YH (2002) Minimizing makespan in a two-machine flowshop with dynamic arrivals allowed. Comput Oper Res 29(3):275–294

21. Ueda K, Hatono I, Fujii N, Vaario J (2000) "Reinforcement learning approaches to biological manufacturing systems". CIRP Ann Manuf Technol 49(1):343–346
22. Ueda K, Markus A, Monostori L, Kals HHJ, Arai T (2001) Emergent synthesis methodologies for manufacturing. CIRP Ann Manuf Technol 50(2):535–551
23. Vaario J, Ueda, K. (1997) An emergent modelling method for dynamic scheduling. J Intell Manuf 9(2):129–140
24. Windt K, Hülsmann M (2007) Changing paradigms in logistics – understanding the shift from conventional control to autonomous cooperation and control. In: Hülsmann M, Windt K (eds) Understanding autonomous cooperation & control – the impact of autonomy on management, information, communication, and material flow. Springer, Berlin, pp 4–16
25. Windt K, Böse F, Philipp T (2008) Autonomy in production logistics – identification, characterisation and application. Int J Robot CIM 24(4):572–578
26. Windt K, Becker T (2009) Applying autonomous control methods in different logistic processes – a comparison by using an autonomous control application matrix. Proceedings of the 17th Mediterranean Conference on Control and Automation. Thessaloniki, Greece