Centralized Rescheduling Approach against Autonomous Control Approach in Dynamic Flexible Flow Shop in the Presence of Machine Breakdowns

A. Mehrsai*[1] and A. Ait Alla*[2]

Dept. of Planning and Control of Production Systems, Bremen University, BIBA Bremen, Germany
*E-Mail: meh@biba.uni-bremen.de[1], ait@biba.uni-bremen.de[2]

Summary: In the current paper, we study the concept of dynamic scheduling in flexible flow Shop with limited carrier under stochastic machine breakdowns and processing time. In most flexible flow shop scheduling methods reported in the literature consider a determinist environment, where all data of the problem are known in advance. Most of these works focused on the development of methods to produce a feasible schedule of good quality in short time computing. However, real systems are stochastic and dynamic such that the initial generated schedule must deal with the presence of a variety of unexpected disruptions that may occur dynamically and cause deviations from the initial schedule. To cope with machine breakdowns and stochastic processing time, a centralized heuristic based on local search simulated annealing algorithm is used to implement the reactive scheduling for the dynamic scheduling problem. This centralized heuristic method will be compared with a decentralized autonomous approach. In order to evaluate the performance of the both developed approaches two performance measures, namely the average flow time and the makespan, were be used.

Keywords: Meta-Heuristic, Autonomous Control, Flexible Flow Shop Scheduling, Simulation.

1. General

In recent years, scheduling problems have become the subjects of many studies about tasks allocation. Depending on availability of jobs prior to a schedule creation, scheduling systems can be classified into static and dynamic ones. In static scheduling, all jobs' specifications are identified before creating the respective schedule and the production sequence does not change during processing. In dynamic scheduling, dynamic events like random job arrivals and machine breakdowns can be appeared, so that they affect the normal execution of the initially generated schedule. Therefore, a rescheduling is necessary to adapt the predefined schedule [1]. Conventionally, most of scheduling systems developed in manufacturing environments are centralized. Indeed, centralized systems provide better schedules, but they tend to face problems in dealing with disturbances in practice [2]. Furthermore, the continuously growing complexity in scheduling problems, inspired by dynamic circumstances, forces schedulers to simplify the centralized problems by decomposing them into decentralized sub-problems with local challenges. Thus, a large portion of scheduling research tries to employ decentralized architecture rather in dynamic scheduling. However, among several mechanisms, autonomous control seems to be well adapted to such problems and architectures. This holds true because resources in a scheduling environment have the ability to generate their own local schedules, to react locally to domestic changes, and to develop a good global schedule through cooperation with each other [11]. Thereby, the complexity can be reduced, while the flexibility and fault tolerance is getting enhanced. The authors in [3] investigated the performance of autonomous control compared to different heuristics in a flexible flow shop (FFS) in the presence of uncertainty caused by stochastic arrival of new jobs. They confirmed the efficiency of autonomous control in dealing with stochastic arrival of new jobs in FFS. Nevertheless, the challenge with machine breakdowns in FFS remains uncovered.

Generally, scheduling problems in manufacturing are categorized in three main types as flow shop, job shop, and open shop [5]. A scheduling problem of size $n\times m$ consists of n jobs $\{J_1, J_2,.., J_n\}$ to be processed on m machines $\{M_1, M_2,.., M_m\}$. Briefly, in the job shop for each J_i a sequence of k_i operations $O_i=(o_{i1}, o_{i2},..,o_{iki})$ exist that configure a certain machine processing order for J_i. This order is called technological constraint for the respective job. The flow shop is a special case of job shop which all jobs have the same processing order O_i. However, the jobs may or may not be identical [4] [5]. Furthermore, each coming job J_i can have its own release time r_i, flow time F_i (starts from the readiness of the job until its final operation), and also completion time C_i of the last operation of the job. A machine can process at most one job at a time and a job can be processed by at most one machine at a time. In generic context FFS is an extension to conventional flow shop problem, where at least one processing stage has more than one machine in parallel. Moreover, pre-emption of processing is not allowed. The problem consists of assigning jobs to machines at each stage and sequencing the jobs assigned to the same machine, so that defined optimality criteria are minimized. Nonetheless, the conventional scheduling problems have some assumptions that usually are not realistic in practice. For instance, there is no arrival of new jobs after the scheduling is done and the number of arriving jobs is pre-known and their demand is received beforehand. However, here the dynamic aspects of scheduling are introduced by machine breakdowns and stochastic processing time.

2. Dynamic Scheduling

2.1 Dynamic scheduling in conventional centralized approach

The problem of dealing with uncertainty and stochastic events is addressed in literature by dynamic scheduling. Three general approaches of scheduling in consideration of uncertainties can be found as proactive scheduling, reactive scheduling, and predictive–reactive scheduling [1] [6].

Proactive scheduling is often used when the uncertainty can be quantified in some ways. It tries to calculate a robust schedule by taking into consideration uncertainties, which could occur during the execution of the schedule. *Reactive scheduling* takes place at the time of execution of the schedule and the decisions can be made locally in real-time. This approach is often used in highly dynamic and complex environments, which are very difficult to control. *Predictive-reactive scheduling* is a hybrid strategy, which consists of two phases. In the first phase "predictive scheduling", a preventive schedule (or so-called baseline) is generated. This baseline schedule determines the start and completion times of jobs based on given requirements and constraints. The second phase "reactive scheduling" is about the execution of the schedule and its revision in response to unexpected real-time events [7].

Since the scheduling problem in complex manufacturing system is strongly NP-hard, the use of heuristic methods like dispatching rules is a common approach. However, in recent years, meta-heuristics (tabu-search, simulated annealing, and genetic algorithm) have been successfully employed to solve dynamic flexible flow shop scheduling problems.

In this paper a genetic algorithm (GA) is used to generate a good initial schedule to assign products to machines by means of product types. A machine breakdown will be introduced to incorporate the disruption in the production system. Once a machine is failed, it cannot perform any job until it will be repaired. Additionally, the fixing time is assumed to be known in the simulation. Indeed, such disruptions need to be mitigated through a rescheduling action. In this regard, simulated annealing (SA) algorithm generates a partial schedule that attempts to revise the initial schedule for responding to the environment changes without rescheduling the complete schedule. In other words, the initial schedule generated by GA has initially defined the best schedule for operations, in which the breakdowns in progressing interrupt it.

2.2 Dynamic scheduling in decentralized autonomous approach

Autonomous control describes processes of decentralized decision-making in heterarchical structures. The objective of autonomous control is to achieve more robustness and positive performance of a total system, due to distributed and flexible sub-systems, challenging with dynamics and complexity [8]. According to this definition, autonomous control is characterized by a shift of decision-making capabilities form a total centralized system to its elements. Generally, elements of an entire system in a manufacturing and logistics environment encompass several objects. These elements may span single parts of products to machines in shop-floors and, in a broader scale, to factories of a supply network. More specifically, in scheduling problems autonomous control approach allows intelligent logistic objects to find their own processing sequences by themselves through a network, concerning their own objectives [9]. Generally, the term of intelligent autonomous objects in logistics covers physical objects (e.g. parts, machines, carriers, etc.), as well as immaterial objects (e.g. production orders). Due to the novel information and communication technologies, these objects are able to interact with each other and to gather information about the current states of local systems. These intelligent logistic objects are able to generate decisions according to their own logistic targets on the basis of this information. This kind of decentralized decision-making may positively influence the system's behavior and helps to improve the handling of dynamics, like occurrence of unforeseen events (e.g. machine breakdowns). One of the enabling methods for autonomous activities is learning, also called recognizing the patterns [10]. In this manner, an autonomous object is able to distinguish and learn the local behaviors, besides, making decisions based on learning by compromising rewards and penalties.

In particular, the considered intelligent objects in this paper are autonomous carriers in flexible shop floor. The autonomous carriers employ fuzzy controller to realize autonomy in their decisions [11]. Indeed, each carrier compares the stochastic waiting times of parallel stations and chooses the best machine with the least waiting time, based on fuzzy sets.

3. Problem description

FFS scheduling can be seen as an extension to the general form of flow shop scheduling problem. Thus, the FFS to be considered in this paper consists of 3 series of production stages, each of which has three identical machines operating in parallel and one (un)load station that produce three types of products, see Fig. 1. All jobs released to the FFS have to visit all the stages in the same order. Table I shows the mean processing time (μ) with normal distribution of each stage pro product types. We consider three types of end products, with a constant number of carriers pro product types, which transport the product to be performed into different stages. Whenever, a product is arrived two possibilities happen, either the respective carrier with the same type of product is available and get released to the system, or any carrier is not available and the product muss wait till a carrier with the same type comes to load station. This limitation in the carrier number makes the problem more complex and a challenging task, since an additional constraint is added into the conventional problem description of FFS. Indeed, in industrial practices with waste elimination approach (like pull system in lean manufacturing) product carriers represent the production capacity of the system. In other words, lean balancing can be occurred by limited number of carriers [11]. We want to reflect this practical performance in our scheduling problem.

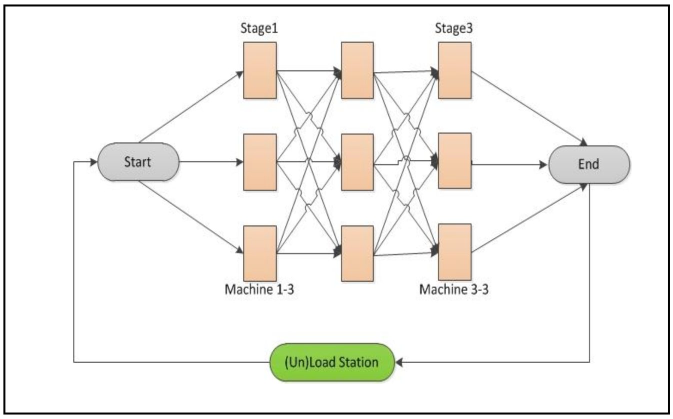

Figure 1. Flexible flow Shop with 3x3 paralleled machines.

Table 1. Products processing times in minutes.

Products Processing Times			
Product Type	Stage		
	1	2	3
1	2:00	2:40	2:20
2	2:20	2:00	2:20
3	2:40	2:20	2:00

3.1 Problem formulation

In this paper, the centralized heuristic system for scheduling is considered as the optimum scheduling for conventional situations. However, this initial schedule is used for the comparison with other alternatives at the presence of breakdowns in both centralized heuristic schedule and autonomous control with decentralized approach. Generally, the experimented scenarios are following:

- Centralized heuristic scheduling once with breakdowns and stochastic processing times (normal distribution) and once without breakdowns and with deterministic processing times.
- Decentralized autonomous control by carriers with breakdowns and stochastic processing times.

We assume that the arrival date of products is known in advance.

The used notations for the considered scheduling problem are as follows.

Indices:
N: number of jobs.
T: number of product types.
S: number of stages.

Parameters and variables:
N^s: number of parallel machine at stage s.
$P_{i,j}$: mean processing time of job j on machine of stage i.
$C_{i,j}$: the completion time of job j on machine of stage i.
C_j : completion time of job j; $C_j = \sum_{i=1}^{s} C_{i,j}$
r_j: the point of time at which the job j is available.
F_j : Flow time of job j; $F_j = (C_j - r_j)$

C_{max}: makespan; $C_{max} = \max_{j=1,\dots,N}\{C_j\}$

F_{ave}: average flow time: $F_{ave} = \sum_{j=1}^{k} \frac{Fj}{k}$

The objective in this paper is just to find the best approach that minimize the makespan of the entire products and minimize the average flow times of all products in every type. Other conflicting objectives are going to be considered in further works.

4. Problem description

Several Simulation runs have been set up in order to analyse the impact of machine breakdowns. The examined variant is 12 pallets (4 each type), 48 products (16 each type). A machine breakdown is introduced each 18 minutes in a machine in stage 2 and a machine in stage 3. The time to repair the machines is 10 minutes in one experiment and 4 minutes in another experiment.

Table 2. Comparison of centralized and decentralized approach.

	Makespan		Average Flow Time	
	Centralized	Autonomous	Centralized	Autonomous
Without machine break-downs	33:13	35:17	08:04	13:02
With machine break-downs (4 Minutes to repair)	36:29	36:22	08:41	13:42
With machine break-downs (10 Minutes to repair)	38:29	45:36	09:18	15:18

References

[1] Vieira, G., Hermann, J., Lin, E., 2003, Rescheduling manufacturing systems: a framework of strategies, policies, and methods. Journal of scheduling, 6/1:39-62.
[2] Ouelhadj, D., Petrovic, S., 2008, A survey of dynamic scheduling in manufacturing systems. Journal of scheduling, 12/4:417-431.
[3] Scholz-Reiter, B.; Rekersbrink, H.; Görges, M., 2010: Dynamic flexible flow shop problems-Scheduling heuristics vs. autonomous control. In: CIRP Annals - Manufacturing Technology, 59/1:465-468.
[4] Wang H., 2005. Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions, Expert Systems 22:78-85.
[5] Pinedo, M., Scheduling: Theory, Algorithms, and Systems. 2008, New York: Springer-Verlag. ISBN: 978-0-387-78934-7.
[6] S. Mehta, R. Uzsoy, 1999, Predictable scheduling of a single machine subject to breakdowns. International Journal of Computer Integrated Manufacturing, 12/1:15–38.
[7] Li, Z., Ierapetritou, M., 2008, Process scheduling under uncertainty: Review and challenges, Computers & Chemical Engineering, 32/4:715-727.
[8] Scholz-Reiter, B.; Jagalski, T., 2008, Dynamics of Autonomous Control in Production Logistics., 26th International Conference of the System Dynamics Society. System Dynamics Society, Albany, USA.
[9] Scholz-Reiter, B.; Görges, M.; Jagalski, T.; Mehrsai, A., 2010, Modelling and Analysis of Autonomously Controlled Production Networks, Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 09):850-855.
[10] Mehrsai, A., Wenning, B. L., Scholz-Reiter, B., 2011, Analysis of learning pallets in flexible scheduling by closed queue network, International Symposium on Assembly and Manufacturing (ISAM), IEEE Explore, Tampere.
[11] Afshin Mehrsai, Hamid-Reza Karimi und Bernd Scholz-Reiter, 2012, Toward learning autonomous pallets by using fuzzy rules, applied in a Conwip system, The International Journal of Advanced Manufacturing Technology (IJAMT):1-20.