© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 6, No. 3, Sep-Dec 2015, pp. 34-48. ISSN: 2007-1558.

Rescheduling in Industrial Environments: Emerging Technologies and Forthcoming Trends

Carmen L. García-Mata [1,2], Pedro R. Márquez-Gutiérrez [2] and Larysa Burtseva [1]

[1] Engineering Institute, Autonomous University of Baja California, Calle de la Normal S/N, Col.Insurgentes Este, Mexicali, B.C., C. P. 21280, México

[2] Technological Institute of Chihuahua Ave. Tecnológico 2909, Chihuahua, Chih., C.P. 31310, México

{clgarcia,pmarquez}@itchihuahua.edu.mx, burtseva@uabc.edu.mx

Abstract. The scheduling problems in Manufacturing Systems are characterized by a high degree of uncertainties arising from diverse factors such as stochastic environments and data incompleteness. However, traditionally the schedules generated for those environments are deterministic, or quasi-deterministic at the best, and only recently the approach static is being shifted for a stochastic approach. This paper highlights the uncertainty characteristics that should be taken into account to improve the schedule robustness. In this research, selected cases from the last ten years of stochastic scheduling literature are reviewed, specially the ones relatives to Semiconductor Factories. Another important objective of our report is to bring the attention of researchers to emerging methodologies and technologies coming from the subject of Knowledge Representation and Reasoning. These new methodologies and technologies are well suited to solve hard combinatorial problems with incomplete knowledge.

Keywords: Dynamic scheduling, scheduling under uncertainty, rescheduling, non-monotonic reasoning, knowledege representation and reasoning, answer set programming.

0 Introduction

There are a lot of situations in industrial environments in which one must make a decision under uncertain conditions. Planning and scheduling are plagued of such problems. Different methods have been proposed and applied to deal with unexpected interruptions during the scheduling execution [46]. However, in spite of existing multiple problem instances of stochastic nature, scheduling under uncertainty has only been addressed in a partial and limited way [3], [53]. A representative case of using this modeling approach is a Semiconductor Manufacturing System (SMS). The SMS represents one of the most stochastic production processes, and yet most of the reported research works about those environments have been formulated deterministically [4],[75].

In manufacturing systems the scheduling performance and the production process are inextricably interlinked. During the schedule execution the environment is constantly changing due to the system evolution over time. There are also other changes of non-deterministic nature, which frequently induce interruptions in the manufacturing process, like restricted capacity of the production system, changes in job orders, etc. Usually, these perturbations make the original schedule non-viable, and a partial or complete rescheduling is required.

Different manufacturing systems share common problems, but each one has very specific particularities derived from the production process and the working environment. Rescheduling in Semiconductor Manufacturing has been recognized by the research community as one of the most difficult problems of this type [14], [84]. The difficulty of the SMS rescheduling comes directly from the production processes, which involves complex product flows, quick changing orders, concurrent manufacturing of distinct products, and multiple steps of variable length in the production cycles, besides of the huge quantity of interruptions related to machine breakdowns, rush orders, failures in supplies and deliveries, and many other disruptions that

Received Jun 19, 2015 / Accepted Dec 17, 2015

cannot be completely anticipated. Additionally to the complex production process, a schedule should satisfy multiple objectives such as a minimum makespan or tardiness [4].

In the last decade, researchers have been forced to include uncertainty in rescheduling models to bridge the gap between traditional theoretical models and real world planning problems. To formulate a problem of scheduling under uncertainty, some design decisions must be taken, e.g., which stochastic events should be attended, which event priority should be used, how often rescheduling should be done, etc. All these choices must be carefully weighted for each particular problem, but in general it is required to define a framework of decisions before the problem is modeled. Given the diversity of application areas and the particular characteristics of each scheduling problem, it is unlikely that all rescheduling problems can be formulated inside a single framework. For the purpose of study, some classification schemes have been proposed, and among them the most influential ones were proposed by [3], [82].

In the case of SMS, some surveys about static scheduling problems have been recently published, see e.g. [58], which is restricted to scheduling wafer fab operations. In the previously mentionated paper, one of the identified challenges is to evaluate the impact of rescheduling strategies in the schedule robustness and stability improvement.

The paradigm shift witnessed over the last decade from a static view of scheduling to one including uncertainties can be explained by two main reasons: a) it is imperative to fill the gap between purely theoretical investigations and scheduling decisions towards solving practical real problems within the industry; b) the production processes, and consequently, the related scheduling problems have become extremely complex.

Up-to-date reviews about the rescheduling problem for manufacturing systems are rather scarce. One of such reviews was given by Li and Ierapetritou [47], which is addressed to scheduling problems related to factories with batch processing. The aim of this review is the scheduling problem under uncertainty for Manufacturing Systems in general, and to rescheduling in SMS in particular. This review is biased to works based on the Artificial Intelligence (AI)'s subtopic known as Knowledge Representation and Reasoning (KRR).

The rest of this paper is organized as follows. In Section 2 we discuss the pros and cons of optimal schedules and robust reschedules. Section 3 is dedicated to the origin of uncertainty and what role should be assigned during the modeling step of scheduling problems. Section 4 is devoted to study how the rescheduling problems can be classified based on strategies, policies and methods taking as reference the most influential rescheduling frameworks proposed by diverse authors. Some rescheduling cases were selected and studied in Section 5 with the aim to identify a possible relationship between the scheduling strategy and the technique used to solve the problem. Lastly, Section 6 briefly reviews some theories and techniques from Knowledge Representation and Reasoning well suited to solve scheduling problems with incomplete knowledge. Some concluding remarks complete the paper.

2 Scheduling Optimality versus Rescheduling Robustness

Traditionally, scheduling research has been mainly focused on two main concerns: optimality and efficiency. The optimal use of time and resources has a direct benefit in reducing production costs, while efficiency is related to computational resources. This means that fast algorithms with low computational load are needed. However, it may not be practical to devote too much effort in achieving optimality, since truthful optimal scheduling can only be ascertained conjoined with its practice on real world industrial problems.

In order to meet industrial needs, researchers have begun to put more emphasis on robustness than optimality. In real-world rescheduling problems, stability is important as optimality and efficiency. Stability is measured as the number of resources we need to reassign to a new schedule whenever the current schedule becomes useless [22], [71].

Under this perspective, it is important to consider that the most tightly optimized schedule is probably not the best one. In other words, if the main objective is to maintain the stability of the system, then the most important point is to be sure that the new plan requires as few resource reallocation as possible. Another related concept to stability is shop floor nervousness. This phenomenon happens when a rescheduling process results in a big number of changes in the resource allocation. Experimental results have shown that *scheduling nervviousness* is an undesirable situation, because the production costs increase and negatively affect the makespan [82].

As it previously stated, research in scheduling is focused on *optimality* and *efficiency* while rescheduling is on *robustness*. Under the classical scheduling view point, optimality is determined only for static environments. Consequently, the schedule can only be optimal if the real world behaves as expected during the schedule execution. For quasi-deterministic environments, it is advantageous to assume that the environment is static, since this leads to more tractable variants [71]. In contrast, there are evidences that show that for environments with high uncertainty, it is more convenient to design a robust schedule instead of an optimized one [19].

Two approaches have been reported in literature for modeling non-deterministic scheduling problems. 1) When modeling the problem, the scheduler uses as much information about uncertainties as there is available. Both, Stochastic Programming [8], [28], and Robust Optimization [71] belong to this approach. 2) The problem is solved by dividing it into two parts: one layer deals with the dynamic nature of the problem, while the other solves the deterministic part. This approach is named interchangeably as Dynamic or Reactive Scheduling [42].

Both approaches have pros and cons. Stochastic Programming and Robust Optimization have reported better results than the second approach, because whenever uncertainties are incorporated, the model is closer to reality. Unfortunately, adding uncertainty to the model is expensive, as the problem's complexity class is increased. The complexity of a scheduling problem without uncertainties belongs to the class of NP-complete problems, while stochastic scheduling belongs to the class of PSPACE-complete problems [63].

The main advantages of the dynamic scheduling approach are: 1) Tractability through splitting the problem into two different entities, the scheduler part and the reactive part, allowing to solve large instances efficiently by using a deterministic solver; 2) Faster problem solutions with deterministic solvers are already available; 3) Simplification of uncertainty modeling because it is not necessary to know the probability distributions of the variables - information that in general is hard to obtain. Disadvantages of the dynamic scheduling approach are: 1) dynamic rescheduling does not incorporate uncertainties during the modeling of the problem and, consequently, the resultant schedule is less robust than the one obtained using the first approach. 2) There are few theoretical results reported to date. Some of the most recent frameworks for reactive rescheduling were proposed in [23], and [42].

3 The Role of Uncertainty in Modeling

In the taxonomy proposed by Tannert et al. (2007), uncertainties are classified into two main groups: objective and subjective. Subjective uncertainty is related to moral aspects and is beyond the scope of this paper.

3.1 Objective Uncertainty

Tannert subdivides objective uncertainty into two subclasses: epistemic and ontological. Epistemic uncertainty is caused by gaps in knowledge about some topics. Ontological uncertainty is derived from the stochastic nature of a particular situation. Usually, complex systems exhibit this type of behavior. These systems often have a non-linear behavior, which requires to solve uncertainties by an inference method different to that of deterministic reasoning.

Regardless of of Tannert's categorization, uncertainty is mainly defined depending on how it is measured. There are different methods to measure uncertainty. For example, to measure risks or the possible occurrence of events, probabilistic uncertainty is used. To measure the grade of certainty and the membership to a set, fuzzy sets and fuzzy logic are used. Meanwhile, the possibility theory is devoted to handling incomplete information and it is similar to probability theory, but differs in some aspects. Possibility Theory uses a pair of functions called possibility and necessity measures. Dubois [17], identifies at least four kinds of ideas that can be taken into the possibility world to get a measure function of these concepts: a) feasibility (if it is possible to solve a problem); b) plausibility (if some events are likely to occur); c) consistency with available information (if a proposition does not contradict the available information), and d) if something is allowed by the law.

In conclusion, as all scientists agree, scientific research is more about probabilities and possibilities than truths beyond all doubt. Uncertainty permeates everything in the real world. Events seemingly inconsequential, can have catastrophic impacts in apparently unrelated facts. It does not matter how detailed a model can be, a researcher has never a complete knowledge of the problem under modeling. So, uncertainty becomes a conceptual challenge, and scheduling researchers should confront it. Undoubtedly, we cannot keep ignoring the fact that the real world is uncertain. Therefore, it is important to anticipate how to react under unexpected events that turn the schedule under execution into a nonviable schedule. Additionally, if probabilistic or possibilistic information about random events is available, or if it is feasible to obtain, it is important to take it into consideration to assess the impact of unexpected events on system performance.

3.2 Scheduling under Uncertainty

As Schrader [73], pointed out in his seminal paper, whenever a problem is being solved, one of the most crucial options to be made is the level of uncertainty and ambiguity that should be incorporated into the model. Schrader remarks that the problem should not be derived without an adequate recognition of the external sources of uncertainty. He clearly distinguishes that during the framework conformation the designer has to carefully choose what kind of uncertainty and ambiguity should be incorporated into the model. These design decisions later determine which mathematical model and methodology are the best, given the chosen characteristics of the framework. Another design issue is to identify what parts of the problem change as the system evolves over time and what parts stay static. Fixing these restrictions helps keep the problem tractable. Besides, if the domain problem is restricted, the search space of the solution is narrowed.

However, the consideration of which and how the uncertainties should be integrated into the model, is not a trivial problem. Aytug et al [3], proposed the uncertainty as the principal actor in the rescheduling process and even put forward a taxonomy for uncertainty. They categorized uncertainty in four preliminary dimensions: cause (object, state); context (free or sensitive);

3.3 Uncertainty Sources in Manufacturing Systems

In production environments, there exists a myriad of uncertainty sources affecting the schedule execution. Some of these uncertainties can be anticipated, and it is possible to take some measures during the scheduling process. Nevertheless, there are other types of uncertainties that are hard to anticipate. In general, for Manufacturing Systems, the disturbances are related to load capacity [28], job orders, or both. According to Katragjini et al [39], the most common disruptions related with load capacity are: a) machine breakdowns (the most studied); b) non-available tools; c) absence of operators; and d) deterioration of machines´ efficiencies. On the other hand, the most common disruptions related to job orders are: a) rush orders; b) priority changes; c) variations in processing times; d) order cancellations; e) rework; and f) lack of material or material transportation delays.

In a SMS, the production process is complex, dynamic and plenty of uncertainties come from different sources. These uncertainties are particularly related to the volatility of the market, which is constantly changing product demands [11]. Other unexpected events are directly related to the production process itself, such as the case of jobs requiring re-working in some of the production steps, or interruptions derived from simultaneous manufacturing of different products.

4 Rescheduling Study Cases: Analysis of Strategies versus Techniques

To model problems of scheduling under uncertainty, the first thing to do is to fix the rescheduling strategies and policies. Different classification schemes for strategies and policies have been proposed [82], [3], and [62], but there is no a general consensus until now about what scheme is the best. The rescheduling frameworks by Vieira and Aytug have common characteristics, but the one of Ouelhadj and Petrovic [62] added a new scheduling strategy to a total of four scheduling strategies: completely reactive, predictive-reactive, robust predictive-reactive, and robust pro-active. We consider that Ouelhadj and Petrovic's taxonomy is more precise because it makes a finer separation among the strategies. That is why we have decided to use it to classify all the systems studied in this review.

In scheduling problems with a total reactive strategy, uncertainty is not taken into account whenever partial schedule is being built, so if an unexpected event ever happens, the schedule is reassessed or re-optimized. In the other three rescheduling strategies, different levels of uncertainty knowledge are embedded into the initial and predictive schedule, and interruptions caused by some unexpected events are attended according to the rescheduling policies previously established. Furthermore, as much as in the case of robust predictive-reactive as in robust pro-active strategies, other additional objectives to be taken into account are robustness and stability.

In the following sections, some study cases are analyzed and classified according to the strategy used to deal with interruptions.

4.1 Reactive Scheduling

Whenever an interruption should be attended in completely reactive scheduling, the actual partial schedule is adjusted based on heuristics or dispatching rules.

Modeling methodologies for SMS's reactive scheduling are more sophisticated than those for simpler manufacturing systems. Researchers hesitate to get a model for a complete Semiconductor Wafer Fabrication System (SWFS) using orthodox mathematical methods. Instead, researchers frequently use alternative modeling techniques, such as languages to describe distributed systems. One popular modeling language for complex systems is Petri nets. Other approaches reported in the literature are Kelly's, Markov's, Brownian, Queing Theory, and Continuous Flow Models, but for SMS, the most frequently scheduling approach is the one based in dispatching rules.

To obtain the information about the actual state of the factory, it is necessary to model the behavior of the system, including one of the most relevant characteristics of SMS', re-entrancy (pattern flow of their manufacturing lines). For example, a scheduling problem for a wafer factory was modeled and simulated through a combination of queuing theory and color-timed Petri nets (CTPN) [84]. The scheduler was implemented by a genetic algorithm that dynamically searches for an appropriate dispatching rule. Experimental results showed that the genetic algorithm-based scheduler has a superior performance compared to conventional dispatching rules. Other authors, such as the ones in [66], report a case of SMS scheduling modeled by hierarchical colored timed Petri nets (HCTPN). They used a combined approach between HCTPN and extended genetic algorithms (EGA) to study how to optimize the combination of scheduling policies. The results obtained by a system simulation showed a near optimal schedule.

In a similar way, in [43] and [49], was used an approach based on time extended object-oriented Petri nets (TEOPNs) for SWFS performance modeling, real-time dispatching and simulation. TEOPNs were used to describe the SWFS as a series of objects. Coincidentally, in both reports by Liu and Lee, dispatching rules were developed via a dynamic bottleneck dispatching algorithm. The performance was evaluated by Liu using a simulation architecture SWFS. Meanwhile, Lee et al. [43] proposed a new multiple-objective scheduling and real-time dispatching (MSRD) approach, which basically consists of two main modules: an off-line multiple-objective scheduling and an on-line real time dispatch. The performance evaluation was done via a simulation built on a platform made of the MSRD prototype and the TEOPNs.

In the studies carried out by Lee and Liu, the virtual SWFS was derived from a real fab located in Shanghai. They obtained similar results showing that a dynamic scheduling dispatch has a better performance than both critical ratio (CR) + First In First Out (FIFO) and Earliest Due Date (EDD) dispatching policies.

A different approximation to modeling the re-entrancy problem in semiconductor's manufacturing is the one reported by Coron et al [13]. These authors characterized the re-entrancy problem as an optimal control problem governed by the scalar hyperbolic conservation law using partial differential equations.

Another methodology that seems well situated for solving complex scheduling problems such as those in SMS, are Multi-Agent Systems (MAS). Many researchers have taken advantage of MAS' superior capacities to deal with randomness and dynamism of complex problems to solve scheduling in manufacturing environments [20], [48], [83] and [90].

For example, Lin et al. [48], developed a distributed simulation platform for a semiconductor manufacturing process. The platform architecture was structured into three layers: The network communication layer, the middleware layer (based on JADE), and the multi-agent simulation layer. This platform was tested simulating a semiconductor manufacturing process of a real semiconductor factory in Shanghai. Meanwhile, Mönch et al. [57] proposed a new architecture of an agent-based system for production control of an SMS, which is implemented as a multi-layer hierarchical scheme.

In order to fulfill the overwhelming demanding requirements of modern production systems and keep the firm competitive against other businesses, production scheduling and execution control should be tightly coupled. This strong integration in the Manufacturing Execution Systems (@MES) is essential to achieve an adaptive behavior due to the interactions between a set of agents acting as autonomous managers, as it happens in the Agent Based Modeling and Simulation (ABMS) tools, proposed by Rolón et al. for a @MES distributed design [69]. Rolon's approach used a bio-inspired technique in Holon Manufacturing Systems [30], [78]. The agents showed emergent behaviors comparable to a complex adaptive system. Besides, the agents are well-suited for modeling processes, where each agent must adapt and modify its own behavior over time. Each agent has autonomy to solve disruptions related to its function. The fulfillment of goals is achieved collaborating with each other, and each agent individually defines and performs his own actions. The agents communicate with each other coordinated by a dynamic Gantt chart. According to the simulation results, the interaction mechanisms among the agents were stable and robust in spite of the total autonomy of all agents and the absence of a master schedule.

4.2 Predictive-Reactive Scheduling

The predictive-reactive strategy was approached in various ways by scheduling researchers, but the initial schedules were designed off-line generally, and whenever unanticipated events occur during the running time, partial or complete rescheduling was done based on rescheduling policies previously established.

Hung et al., used the predictive-reactive strategy for a scheduling problem in the photolitography area of a semiconductor wafer factory [38]. His goal was to compare the effectiveness and efficiency of three algorithms: simulated annealing, genetic algorithm and tabu search to get an optimal reschedule. They also proposed a sensitivity search method to improve the performance. Their method worked as follows: independently of the moment when rescheduling is required, the initial schedule is used as the starting point to search for a new schedule. The experimental results showed that a sensitive search improved the performance, and in particular, tabu search turned out to be superior than the other proposed algorithms.

Recently, new theories and methodologies are being introduced to represent the domain knowledge for rescheduling problems. Muñoz et al., reported a study case found in a chemical factory with multi-product batches [59]. The unexpected event considered is the increasing operation time. The considered objective function is the maximization of the profit of the plant taking into account income and energy costs. The modeling of the problem was based on approximated dynamic models and ontologies. Usually, online and historical information among different decision levels is independent and not properly integrated during the process of rescheduling. However, ontologies can be used to integrate this information as it was done in this study case.

Mixed-line production is trendy in certain types of current production systems and emerged as a way to meet the market demand. Under this production scheme, a customer order consists of high variety and low volume products. Typical manufacturing systems with mixed-line production are the electronics and wireless communication industries. Huang et al., solved a scheduling problem for these kind of manufacturing systems, adopting the Drum-Buffer-Rope (DBR) technique of Theory of Constraints (TOC) and the buffer management [37]. The application of these techniques helps the managers detect early production problems, and to evaluate the desirability of rescheduling in advance. The evaluation objective was to compare the DBR and EDD techniques, and to study the impact of delayed orders on the performance, maximum tardiness time and cost as well as the total completion flow. The results showed that DBR was better than EDD because it minimized the longest tardiness in the customer's orders, and it offered a greater flexibility and ability to manage a fab with capacity overload and frequent disruptions.

4.3 Robust Predictive-Reactive Scheduling

Scheduling of highly stochastic systems is a hard problem because they present unpredictable behaviors. However, it is not convenient to reschedule at each disruption either. Instead of a policy of rescheduling triggered every time a disturbance happens, a better option is to reschedule just whenever the interruption dimension makes the running schedule non-viable [12]. A generic and optimal solution for the robust predictive-reactive scheduling problem is still an open question, and multiple approaches and methods are constantly being proposed and tested.

A research by Vonder et al., reported the results of an experiment designed to evaluate several predictive reactive resource-constrained project schedules [80]. The projects of this kind are stochastic versions of the basic scheduling problem in a deterministic setting, known as a resource-constrained project schedule problem (RCPSP). In the extended RCPSP schedule, stochastic activity durations are considered, and its optimization objective is to minimize the expected makespan. Another objective is to guarantee that the schedule robustness should not be affected by disturbances. Vonder's complete experiment consisted in evaluating all possible combinations of three baseline schedules obtained for different procedures with four reactive rescheduling procedures. The impact on the performance is measured whenever there are range variations in the following parameters: 1) the level of uncertainty in the activity span; 2) the weighting parameter (ratio of the dummy end activity to the average of the rest of activities); and, 3) the project due date (timely project completion probability, TPCP).

The baseline scheduling methods evaluated were: 1) RCPSP-predictive: an exact procedure with the average duration of the activities; 2) a suboptimal procedure based on simple priority-based scheduling heuristics, specifically, on the Latest Start Time (LST) priority rule; 3) Resource Flow Dependent Float Factor (RFDFF): suboptimal procedures targeted to minimize the stability cost function. The four reactive scheduling methods are: 1) RCPSP-reactive: complete rescheduling by an exact procedure, where only the finished activities at the disruption time are considered for the rescheduling; 2) Fix Flow: no complete rescheduling under the railway concept, that is, activities never start earlier than its assigned starting point on the base schedule; 3) activity-based priority rules (ABR): the problem is solved by heuristics, and the solution is an activity list rather than a schedule; 4) resource-constrained earliness-tardiness project scheduling problem (RCPSPWET): using exact procedures earliness-tardiness costs are minimized.

The final conclusion of this experiment was that even when exact procedures were used to generate proactive and reactive schedules, the TPCP's optimization objective showed the best results. However, the stability objective was not improved. In general, considering the results obtained in all the experiments, the authors arrived to the conclusion that for strong requirements in TPCP, not too tight due dates, and low values of the duration variability, it is better to generate a robust (stable) proactive scheduling based on RFDFF heuristics. However, for highly variable environments and TPCP with low values, the RFDFF heuristic is not the best option. In these cases, a better option is to combine a procedure that generates a minimum duration baseline schedule with a stability-improving reactive policy, like WET. In conclusion, the authors adviced to conduct more studies about robust reactive scheduling in order to improve the WET results.

Kuster et al., proposed a generic approach to partial scheduling in highly stochastic realistic environments [41]. First, they proposed an extension of the conceptual framework RCPSP, (x-RCPSP) to describe formally disturbance management problems. In essence, x-RCPSP conceptualized *active* and *inactive* elements. Only *active* elements were considered for rescheduling. The authors also proposed a Local Rescheduling (LRS) to do the partial scheduling. LRS was based on a time window that is extended in a bidirectional way to search for potential solutions. These potential solutions must fulfill the new requirements imposed by the occurrence of stochastic events.

A similar approach and objectives were addressed by Huang et al., in their Job Shop Scheduling Repair (JSSR) research problem [36]. The goal of this problem was to obtain a stable repair scheduling through a commitment between makespan optimization and performance deviation during rescheduling. The problem was formulated as a Disjunctive Temporal Problem (DTP), framed as an Optimal Constraint Satisfaction Problem (OCSP) and solved by an algorithm integrating incremental consistency and efficient candidate generation.

4.4 Robust Pro-Active Scheduling

In this strategy, the objective is to obtain a robust and stable schedule. To accomplish this goal, robust and stable schedules are created off-line including uncertainties. In theory, the resulting schedule should be insensitive to disruptions. However, unavoidably some non-anticipated exogenous events occur during the execution of the proactive schedule. These disturbances frequently cause the schedule to be partially or completely repaired. The new schedule should be generated without missing that stability and performance should not decrease.

Robust proactive approaches are classified into three subcategories: 1) redundancy-based, 2) probabilistic, and 3) contingent/policy-based techniques. The objectives of each one are: 1) to reduce the impact of uncertainties by allocating time and extra resources; 2) to obtain the probability density functions of uncertainties, and; 3) to establish policies of scheduling to attend any particular sequence of events [52].

Besides the stability measures, it is also important to know how the disturbances and the rescheduling policies affect the system performance. Robust schedules allow us to correlate the number of disturbances with the system performance. Diverse studies have been developed to measure how disruptions affect the system performance [29] and [40].

Bonfill et al., proposed a stochastic modeling and optimization approach to solve a rescheduling problem of batch processing [9]. Initially, a proactive schedule is generated including uncertainty measures of loading, heating and discharging. The uncertainties are characterized by a uniform distribution. The optimization objective is a combination of makespan and waiting times. The schedule is obtained using an optimized genetic algorithm. Whenever a machine breaks or the processing times are longer than expected, a new schedule is calculated applying the right-shift rule. Finally, the authors developed an experiment to compare the performance of the algorithms under deterministic and stochastic approaches. They found that even when the makespan and the waiting times were optimized for a deterministic environment, under a stochastic scenario the makespan increases by about 4%. Despite the simplicity of this problem, the experimental results showed significant information supporting benefits if uncertainties are incorporated from the beginning into the schedule.

Merdan et al., studied the case when an event-driven rescheduling policy was applied to a small manufacturing system with machine failure as the triggering event of rescheduling [56]. The system performance was empirically evaluated for different failure types with diverse spans as well. The influence of the number of pallets over the system performance was also evaluated, and several rescheduling methodologies were tested, such as Right-shift Scheduling (RS), Agenda Rerouting (AR), New jobs Rerouting (NR), and Complete Rerouting (CR). The evaluation was implemented on the agent-based simulation environment MAST. The results showed that CR was the best rescheduling methodology in this particular case.

In the research reported in [86], a predictive-reactive rescheduling with non-reshuffle and reshuffle strategies was proposed for a flexible manufacturing system (FMS). The authors considered new job orders arriving when the scheduled jobs were not finished as the only source of disturbance. The challenge was to integrate the new job orders into the existing production schedule immediately, while preserving factory performance and stability. In the non-reshuffling strategy new orders are assigned to machines just in the available idle times, while in the reshuffle strategy, operations are re-sequenced to generate a partial solution within the rescheduling horizon. The performance measures are a commitment between the sum of weighted squared tardiness, the makespan and the stability. The stability was calculated based on three aspects: machine migration, job start time, and sequence deviations. The implementations of reshuffling and non-reshuffling were done with Genetic Algorithms. Their experimental results showed that the non-reshuffle strategy is a better option than the reshuffle strategy because it improved the sum of weighted squared tardiness, and at the same time, stability was highly increased without increasing the cost of makespan.

Other researchers [61], looking for a balance between pros and cons of different kinds of policies and methods, have proposed a framework including the best of all worlds. Novas' framework is oriented towards multi-product and multi-stage plants. The operational policies are of three types: a) unlimited intermediate storage (UIS); b) non-intermediate storage, unlimited wait (NIS-UW); and c) non-intermediate storage, zero-wait (NIS-ZW). The knowledge about the manufacturing environment and the production plan is explicitly represented and modeled with object-oriented techniques. Certain static information about the resources (for example, about the properties and methods of the most relevant entities), is included into the domain knowledge. The temporal attributes of resources are considered as well. The chosen scheduling policy is event-driven together with a partial rescheduling method. The goal of this proposal is to give an immediate response to events without introducing excessive changes into the schedule and at the same time, to maintain the system stability. By means of the domain representation, the current state of the scheduling in process can be known whenever an unanticipated event occurs. Thus, at any moment an event happens, the context can be obtained with precision and used to render the rescheduling problem specification. Otherwise, the incorporation of contextual information could allow a more precise evaluation of the impact of an event. However, the authors proposed this improvement for the future. Contextual information is only used to evaluate whether the schedule becomes useless by the effect of an occurrence of some event, and to decide if rescheduling proceeds. Lastly, once the rescheduling is completely specified and the performance measures have been selected, the model is generated through the Constraint Programming (CP) updating module.

Novas' approach has the advantage of reducing nervousness in the production line and at same time to maintain an acceptable optimization. The use of contextual information demonstrates that, to evaluate the impact of the event allows us to distinguish the cases where the size of the disturbance mandates a rescheduling, (reducing the cases needing rescheduling) avoiding unnecessary rescheduling. These experimental results also demonstrated that better optimization objectives were obtained when the domain knowledge included larger sections of the manufacturing process.

Current manufacturing systems should be capable of re-configuration, flexibility and robustness. Some of those emergent manufacturing paradigms have been inspired by biology, for example, Bionic Manufacturing Systems (BMS), Holonic Manufacturing Systems (HMS) and Reconfigurable Manufacturing Systems. According to the Bio-inspired Paradigms, the key to achieve adaptability and robustness in changing environments is self-organization. Even when the theoretical bases of Bio-inspired paradigms date from mid 60s, only few applications to the industry have adopted this paradigm [44] and [45].

5 Methods Based in Knowledge Representation and Reasoning (KRR)

The next sections are dedicated to a brief overview of the most influential and successful theories and methods based on KRR well suited to the solution of combinatorial search problems, including scheduling/rescheduling and planning problems. The methods and theories selected are: Satisfiability (SAT) solvers, CP, Nonmonotonic Logic, and Answer Set Programming (ASP).

5.1 Combinatorial Search Problems and KRR

Scheduling, Combinatorial Optimization, SAT, the Constraint Satisfaction Problem (CSP), the Quantified Boolean Formula (QBF) satisfaction problem, and Planning (bounded length) all belong to the Combinatorial Search Problems class. The search space of solutions is usually exponential in the size of the input and its complexity is NP-complete for all these problems, except for QBF which is PSPACE-complete.

There are some basic techniques for solving combinatorial search problems ranging from AI to numerical analysis and operations research. However, given the complexity of the problem, a combination of heuristics and combinatorial search methods are usually used to solve the problem in reasonable time.

Although AI techniques such as Fuzzy Logic, Neural Networks and Genetic Algorithms have been used together with many other different methods to solve scheduling problems, there is a set of powerful KRR methods and techniques that have been little explored in solving manufacturing scheduling problems. Nowadays KRR is one of the hot topics in AI. The origins of KRR can be traced back to the seminal papers of John McCarthy [54].

5.2 SAT solvers

A SAT problem is represented as a Boolean equation, and given a certain assignment of values to the variables, the formula evaluates to true, or responds that no such assignment exists. On the other side, it has been determined that AI planning problems are PSPACE-complete, but if the planning problem is restricted to plans of a polynomial size, the reasoning problem turns into an NP-complete problem, which can be efficiently solved on a SAT engine by encoding problem instances in Conjunctive Normal Form (CNF).

SAT solvers are remarkable successful, and have been applied to very diverse problems, including planning and scheduling. An important advantage of SAT is that there are numerous efficient solvers [32]. Although the complexity of SAT is an exponential run time for all known algorithms in the worst-case, experimental results have showed that most real-world problems can be solved by SAT solvers in polynomial time [87] and [89].

5.3 Constraint Programming

The basic idea of CP is to define a problem as a set of constraints represented through a set of variables. A domain of values is defined for each variable, and relationships among the subsets of these variables are established. For example, in a scheduling problem for a manufacturing system, the decision variables could be the batch sizes of each demand and the start times of the jobs. The constraints could be the machine load capacity (only one job can be scheduled to be processed at a time on one machine); and the job sequence order (jobs cannot be scheduled in arbitrary order, but processed according to the production specs).

Recently, the Constraint Satisfaction Problem (CSP) has been extended for different purposes. In some CSP extensions, the objective is to search for optimal solutions of single- and multi-criteria problems. The constraint solvers search the solution space in a local or systematic way. The solvers based on systematic searching use backtracking, branch and bound, or a combination of search and inference. Inference is used to narrow the search space. On the other hand, the non-systematic solvers are based on local search and the solutions are not always optimal because the search is incomplete.

Like SAT, CSP is NP-complete. Fortunately, it is frequent that instances of real-world combinatorial optimization problems show structural properties that can be exploited to design polynomial time algorithms. Some of these structural properties are static while others are dynamic. One static property is the degree of acyclicity in constraint graphs. For example, an instance of a combinatorial optimization problem is solvable in polynomial time if the tree width of its constraint graph is bounded by a constant [72]. On the other side, the approach based on dynamic structural properties focuses on looking for hidden structures in the runtime distribution of the search methods. These hidden structures are known as backdoor sets. Basically, a backdoor set consists of the variables whose instantiation transforms the NP-complete problem to a tractable one [15], [31], and [81].

Unfortunately, there are also some drawbacks in applying CP to the solution of real-world problems. One of them is that it is not easy to find a good model given the chosen solver, so, hunting for a solution is a very hard problem. Another big drawback is that frequently real-world problems are over-constrained and no solution exists for that set of constraints. The researchers circumvent this difficulty in two ways; using preferences instead of constraints or slightly modifying the constraints so that a solution can be found without modifying the original problem too much [16] and [35].

In spite of these inconveniences, in recent years CP has been used alone or together with other methods to solve some scheduling/rescheduling problems in manufacturing systems [50] and [61]. Another case where CP was applied to solve a rescheduling problem in SMS was given in [88].

5.4 Non-monotonic Logic and ASP

The research community in KRR is wide and has addressed multiple applications. KRR's researchers have faced their studies from very different angles. There are research groups focusing on fundamental issues common to different applications, and consequently, their studies have mainly addressed KRR's general methods. Other groups have the objective of developing specialized methods of KRR to handle core domains, such as time, space, causation and action. A third group of researchers have put their attention on the solution of practical applications through KRR. Planning and scheduling are classes among these applications.

The general methods proposed until now have common points but differ in important questions. One of the most important differences comes from the type of monotonic or non-monotonic logic based on. Monotonic logic is most commonly known as classical logic. Unfortunately, classical logic is insufficiently expressive and impractical to deal with incomplete and counterfactual KRR. So far, in real-world domains, rational entities (humans or based on AI) most of the time make inferences in situations where knowledge is incomplete and some facts are contradictory. Therefore, a lot of effort has been invested to extend the classical logic into a new logic capable of handling qualitative and uncertain information.

Reasoning systems based on monotonic logic are incapable to adjust their conclusions whenever new and contradictory facts are added to the preexisting knowledge base. In contrast, the distinctive feature of reasoning systems based on non-monotonic logic is their capability to arrive to consistent inferences even in the presence of counterfactual knowledge.

In the methods based on non-monotonic reasoning, there are two main and successful semantic solutions: stable model semantics [24] and well-founded semantics [79]. Both semantics have numerous extensions.

Answer Set Programming (ASP) is a declarative language for KRR based on stable model semantics [24] and [25]. It is a paradigm designed to solve combinatorial search problems and their optimization variants [7]. The most outstanding characteristic of ASP is its ability to represent defaults, which makes ASP so different from other languages for knowledge representation. Default reasoning is equivalent to common sense reasoning, which is a way humans deal with uncertainties in the environment. In fact, decision-making in the real world is always done under incomplete knowledge. Knowledge gaps are filled based on the assumption “that things go as usually do”, or more formally, “elements of class C normally satisfy property P”.

For example, when we model a scheduling problem for a manufacturing system with non-monotonic logic, it is possible to represent the Epistemic Uncertainty assuming that the machines assigned to each job will work “normally” during the schedule execution, where “normally” means that the machine will not suspend its operation execution for any reason. If a machine is broken or becomes unavailable before the order is completely processed, the scheduler would need to retreat the previous inference and adjust its conclusions according to new and possible contradictory facts.

Although it is known that finding the answer set (solution) to a problem modeled with ASP is NP-complete, it is important to highlight that it is possible to represent all NP-search problems by means of ASP. In other words, expressiveness of ASP makes it possible to represent every property of a finite structure in a precise mathematical sense using first-order structures without any kind of functions. Problem representations in ASP are decidable in nondeterministic polynomial time with an oracle in NP [6].

It is important to refer to ASP's real-world applications, which widely range. We can find among the most remarkable ones: decision support system for a space shuttle [60], configuration [74], scheduling and team generation for a seaport [68], and phylogenetic systematics [18]. No less important it is to highlight that continually, the research community is proposing new extensions of ASP [27], and it is also constantly looking for combinatorial optimization problems that are computationally challenging, in order to propose techniques to solve them in a simple and elegant way by using ASP [75].

6 Conclusions

Recent publications in scheduling testify a paradigm shift from a simplistic view of the world, where uncertainty is exorcised, to a closer view to reality, and every day more and more researchers formulate a scheduling problem in manufacturing systems as a dynamic and non-deterministic one.

For more complex manufacturing systems such as semiconductor manufacturing factories, the task is even more challenging. According to the analyses conducted by the experts from industry and academia, the problem has been approached from a single level of decision. However, to be competitive, SMS requires the integration of more decision levels. Two of these levels relate to the supply chain and real-time plant control [10] and [64].

The modeling of scheduling problems based on dynamic and stochastic systems requires different theories and methodologies than those used for static scheduling. However, a classic mathematical modeling of rescheduling problems in complex manufacturing systems is highly difficult, and alternative modeling methods have been explored. MAS is considered as one of the most promising modeling methods to solve real-world scheduling problems, but it has the disadvantage that causal relations between the variables are not clearly defined. Now, scheduling problems for SMS are not only typical dynamic and stochastic problems, but also have higher levels of difficulty because some jobs have to be re-worked. A resource competition caused by re-working may lead to deadlock situations, consequently, modeling methodologies for rescheduling with probable deadlock should be capable of generating deadlock-free models.

The execution of a scheduling has been modeled as a work-flow problem by different methods; some of them are graphical such as UML or Petri Nets. Diagrammatic modeling has the advantage that it allows an explicit specification of causal relationships between the variables, but it presents issues in the solution validation [65].

Some researchers have explored how to combine the strength of Petri nets and ASP technologies to maximize the capabilities of both technologies to model and solve problems in the area of discrete-event dynamic systems. It is advantageous to use Petri nets due to their ability to represent features such as precedence, concurrency, conflict and synchronization of dynamic systems. Other analysis methods such as structural analysis and graph reachability can be used to prevent deadlock situations. On the other hand, ASP is one of the best options to process algebraic, temporal logic and mathematical equations. These capacities are needed to implement Petri Nets. Another point in favor of ASP is that there exists efficient solvers [2] and [34].

Probably, the solution of a complex scheduling problem with disruptions and changing environment over the time cannot depend on a single method, and hybrid modeling methodologies should be used. KRR's emerging methodologies have many advantages over traditional Operations Research methodologies and meta-heuristics based on AI employed until now in combination with other methods. It is also important to point out that notable advances have been achieved in KRR in the last two decades, both in its theoretical part and in engineering software [33]. Although more research in real-world industrial applications needs to be done, promising results obtained so far in applications based on KRR methodologies open a window for the opportunity of finding a solution to difficult combinatorial problems, such as scheduling/rescheduling problems for real-world manufacturing systems.

Additional advantages of using KRR methodologies for the solution of real-world scheduling problems are twofold: a) knowledge bases can be verified and validated using formal methods [1], and b) it is possible to detect consistency in the reasoning process by formal methods specifically designed for this purpose [67].

References

1. Antoniou, G.: Verification and correctness issues for nonmonotonic knowledge bases, International Journal of Intelligent Systems, 12(10), 725-738 (1997)..

2. Anwar, S., Baral, C., Inoue, K.: Encoding higher level extensions of petri nets in answer set programming. In: LPNMR 2013, Proceedings of the 13th International Conference on Logic Programming and Nonmonotonic Reasoning, (pp.116-121), Corunna, Spain, (Septembre 2013).

3. Aytug, H., Lawley M.A., McKay, K., Mohan, S. and Uzsoy, R.: Executing production schedules in the face of uncertainties: a review and some future directions, European Journal of Operational Research, 161(1), 86-110 (2005).

4. Baez-Sentíes, O., Azzaro-Pantel, C., Pibouleau, L. and Domenech, S.: Multi-objective scheduling for semiconductor manufacturing plants, Computers and Chemical Engineering, 34(4), 555-566 (2010).

5. Balduccini, M. and Gelfond, M.: Logic programs with consistency-restoring rules, in: Doherty, P., McCarthy, J. and Williams, M.A. (Eds.), International Symposium on Logical Formalization of Commonsense Reasoning, pp.9 - 18.Bonatti, P., Calimeri, F., Leone, N. and Ricca F. (2010). Answer set programming, in Dovier, A. and Pontelli, E. (Eds.), A 25-Year Perspective on Logic Programming (pp.159-182). Springer-Verlag Berlin, Heidelberg (2003).

6. Brewka, G., Niemelä, I. and Truszczynski, M.: Answer set optimization. In: IJCAI 2003, Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp.867-872. Acapulco, México (2003).

7. Brcic, M., Kalpic, D., Fertalj, K.: Resource constrained project scheduling under uncertainty: a survey. In: CECIIS 2012, Proceedings of the 23rd Central European Conference on Information and Intelligent Systems, (pp.401 – 409). Varazdin, Croatia (2012),

8. Bonfill, A., Camarasa, A.E., Puigjaner, L.: Proactive approach to address the uncertainty in short-term scheduling, Computers & Chemical Engineering, 32(8), 1689-1706 (2008).

9. Chien, C., Dauzère-Pérès, S., Ehm, H., Fowler, J.W., Jiang, Z., Krishnaswamy, S., Mönch, L. and Uzsoy, R.: Modeling and analysis of semiconductor manufacturing in a shrinking world: challenges and successes. In: WSC 2008, Proceedings of the Winter Simulation Conference, (pp.2093-2099). Miami, FL, USA (2008).

10. Chien, C., Wu, C. and Chiang Y.: Coordinated capacity migration and expansion planning for semiconductor manufacturing under demand uncertainties, International Journal of Production Economics, 135(2), 860-869 (2012).

11. Church, L. and Uzsoy, R.: Analysis of periodic and event-driven rescheduling policies in dynamic shops, International Journal of Computer Integrated Manufacturing, 5(3), 153-163 (1992).

12. Coron, J.M., Kawski, M., Zhiqiang, W.: Analyisis of a conservation law modeling a highly re- entrant manufacturing system, Journal of Industrial and Management Optimization, 14(4), 1337-1359 (2010).

13. Davenport, A.: Integrated maintenance scheduling for semiconductor manufacturing. In: CPAIOR 2010, Proceedings of the International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming for Combinatorial Optimization Problems, Vol. 6140 of LNCS, (pp.92-96), Bologna, Italy (2010).

14. Dilkina, B., Gomes, C. and Sabharwal, A.: Tradeoffs in the complexity of backdoor detection for combinatorial problems. In: CP-07, 13th International Conference on Principles and Practice of Constraint Programming, Vol. 4741 of LNCS, (pp. 256-270), Providence, RI, USA (2007).

15. Downing, N., Feydy T. and Stuckey P.: Unsatisfiable cores for constraint programming. CoRR arxiv.org/abs/1305.1690. (2013) Accessed 16 May 2014.

16. Dubois, D.: Possibility theory and statistical reasoning, Computational Statistics and Data Analysis Journal, 51(1), 47-69 (2006).

17. Erdem, E.: Applications of answer set programming in phylogenetic systematics. In Balduccini M. and Son T.C. (Eds.): Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday, Vol. 6565 of LNCS, pp.415 -431 (2011).

18. Feng, W., Zheng, L. and Li, J.: The robustness of scheduling policies in multi-product manufacturing systems with sequence-dependent setup times and finite buffers, Journal of Computers and Industrial Engineering, 63(4), 1145-1153 (2012).

19. Fitkov-Norris, E.: Literature overview of approaches for enterprise-wide modelling, simulation and optimization. http://eprints.kingston.ac.uk/23969/. (2010) Accessed 10 February 2014.

20. Framiñan, J.M. and Ruiz, R.: Architecture of manufacturing scheduling systems: literature review and an integrated proposal, European Journal of Operational Research, 205(2), 237-246 (2010).

21. Gabrel, V., Murat, C. and Thièle A.: Recent advances in robust optimization and robustness: an overview, European Journal of Operational Research, 235(3), 471- 483 (2014).

22. Gebser, M., Grote, T., Kaminski, R., Obermeier, P., Sabuncu, O. and Schaub, T.: 'Stream reasoning with answer set programming: Preliminary Report. In: KR 2012, Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning, AAAI Press, pp.613-617, Rome, Italy (2012).

23. Gelfond, M, and Lifschitz, V.: The stable model semantics for logic programming. In: ICLP 1988, Proceedings of the Fifth International Conference on Logic Programming, pp.1070-1080, Seattle, Washington, USA (1988).

24. Gelfond, M. and Lifschitz, V.: Classical negation in logic programs and disjunctive databases, New Generation Computing, 9(3–4), 365-386 (1991).

25. Gelfond, M. and Lifschitz, V.: Representing Actions and Change by Logic Programs, Journal of Logic Programming, 17(2,3,4), 301-323 (1993).

26. Gelfond, M. and Yuanlin Zhang, Y.: Vicious circle principle and logic programs with aggregates, Theory and Practice of Logic Programming, 14(4,5), 587-601 (2014).

27. Geng, N., Jiang, Z. and Chen, F.: Stochastic programming based capacity planning for semiconductor wafer fab with uncertain demand and capacity, European Journal of Operational Research, 198(3), 899-908 (2009).

28. Ghezail, F., Pierreval, H. and Hajri-Gabouj, S.: Analysis of robustness in proactive scheduling: a graphical approach, Computers & Industrial Engineering, 58(2), 193-198 (2010).

29. Giret, A. and Botti, V.: Holons and agents, Journal of Intelligent Manufacturing, 15(5), 645-659 (2004).

30. Gómes, C.P., Selman, B., Crato, N., Kautz, H.A.: Heavy-tailed phenomena in satisfiability and constraint satisfaction problems, Journal of Automated Reasoning, 24(1), 67-100 (2000).

31. Gómes, C. P., Kautz, H., Sabharwal, A. and Selman B.: Satisfiability solvers. In Van Harmelen, F., Lifschitz, V. and Porter B. (Eds.): Handbook of Knowledge Representation (Foundations of Artificial Intelligence), Elsevier, (pp. 89-134). Amsterdam, The Netherlands (2008).

32. Harmelen, F. V., Lifschitz, V. and Porter, B.: Handbook of Knowledge Representation (Foundations of Artificial Intelligence), Elsevier, Amsterdam, The Netherlands (2008).

33. Heljanko, K. and Niemelä, I.: Petri net analysis and nonmonotomic reasoning, Lecture Notes in Computer Science, pp. 7-19 (2000).

34. Hoeve, W.: Over-constrained problems. In Milano, M. and Van Hentenryck, P. (Eds.): Hybrid Optimization: the 10 Years of CPAIOR, pp.191-225, Springer (2011).

35. Huang, Y., Zheng, L., Williams, B., Tang, L. and Yang H.: Incremental temporal reasoning in job shop scheduling repair. In: IEEM 2010, Proceedings of the International Conference on Industrial Engineering and Engineering Management, pp.1276-1280. Macau, China (2010).

36. Huang, H.H., Pei, W., Wub, H.H. and May, M.D.: A research on problems of mixed-line production and the re-scheduling, Journal of Robotics and Computer-Integrated Manufacturing, 29(3), 64-72 (2013).

37. Hung, Y.-F., Liang, C.-H., Chen, J. C.: Sensitivity search for the rescheduling of semiconductor photolithography operations, The International Journal of Advanced Manufacturing Technology, 67(1-4), 73-84 (2013).

38. Katragjini, K., Vallada, E., Ruiz, R.: Flowshop rescheduling under different types of disruption, The International Journal of Production Research, 50(1), 780-797 (2013).

39. Kumar, V., Kumar, S., Tiwari, M.K., Chan, F.T.S.: Performance evaluation of FMS under uncertain and dynamic situations, Journal of Engineering Manufacture, 222(7), 915-934 (2008).

40. Kuster, J., Jannach, D. and Friedrich, G.: Applying local rescheduling in response to schedule disruptions, Annals of Operations Research, 180(1), 265-282 (2010).

41. Larsen, R. and Pranzo, M.: A framework for dynamic rescheduling problems. Technical report, Dip. Ingegneria dellInformazione, University of Siena, Siena. www.dii.unisi.it/priv/papers/papers_doc/55.pdf. (2012) Accessed 13 September 2012.

42. Lee, Y.F., Jiang, Z.B. and Liu, H.R.: Multiple-objective scheduling and real-time dispatching for the semiconductor manufacturing system, Computers & Operations Research, 36(3), 866-884 (2009).

43. Leitão, P.: Self-organization in manufacturing systems: challenges and opportunities. In: SASOW 2008, Proceedings of the 2nd. International Conference on Self-Adaptive and Self-Organizing Systems Workshops, IEEE, pp.174-179.Venice, Italy (2008).

44. Leitão, P. and Vrba, P.: Recent developments and future trends of industrial agents. In HoloMAS 2011: Proceedings of the 5th Holonic and Multi-Agent Systems for Manufacturing, Vol. 6867 of LNCS, 15-28. Toulouse, France (2011).

45. Leung, J.Y.T.: Handbook of Scheduling: Algorithms, Models, and Performance Analysis, 1st ed., Chapman & Hall/CRC, London (2004).

46. Li, Z. and Ierapetritou, M.G.: Process scheduling under uncertainty: review and challenges, Computers & Chemical Engineering, 32(4), 715-727 (2008).

47. Lin, J. and Long, Q.: Development of a multi-agent-based distributed simulation platform for semiconductor manufacturing, Expert Systems with Applications, 38(5), 5231-5239 (2011).

48. Liu, H., Jiang, Z. and Fung, R.: Performance modeling, real-time dispatching and simulation of wafer fabrication systems using timed extended object-oriented petri nets, Computers & Industrial Engineering, 56(1), 121-137 (2009) .

49. Liu, S. and Shih, K. Ch.: Construction rescheduling based on a manufacturing rescheduling framework, Automation in Construction, 18(6), 715-723 (2009).

50. Lombardi, M. and Milano, M.: Optimal Methods for Resource Allocation and Scheduling: a Cross-Disciplinary Survey, Constraints, 17(1), 51-85 (2012).

51. Lou, P., Liu, Q., Zhou, Z. and Wang, H., Sun, S.X.: Multi-agent-based proactive–reactive scheduling for a job shop, International J,ournal of Advanced Manufacturing Technology, 59(1 – 4), 311-324 (2012).

52. Ma, Y., Chu, C. and Zuo, C.: A Survey of scheduling with deterministic machine availability constraints, Computers and Industrial Engineering, 58(2), 199-211 (2010).

53. McCarthy, J.: Programs with commonsense. In Proceedings of the Symposium on the Mechanization of Thought Processes, (pp.1–15).National Physiology Lab, Teddington, England (1959).

54. Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I. and Fahl, M.: State-of-the-art review of optimization methods for short-term scheduling of batch processes, Computers and Chemical Engineering, 30(6-7), 913-946 (2006).

55. Merdan, M., Moser, T., Vrba, P. and Biffl, S.: Investigating the robustness of re-scheduling policies with multi-agent system simulation, International Journal of Advanced Manufacturing Technology, 55(1), 355-367 (2011).

56. Mönch, L., Stehli, M., and Zimmermannh, J.: FABMAS: An Agent-based system for production control of semiconductor manufacturing processes. Holonic and Multi-Agent Systems for Manufacturing. Lecture Notes in Computer Science, Vol. 2744, 258-267 (2003).

57. Mönch, L., Lars, L., Fowler, J.W., Dauzère-Pérès, S., Mason, S.J. and Rose, O.: A Survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations, Journal of Scheduling, 14(6)), 583-599 (2011).

58. Muñoz, E., Capón-García, E., Moreno-Benito, M., Espuña, A. and Puigjaner, L.: Scheduling and control decision-making under an integrated information environment, Computers & Chemical Engineering, 35(5), 774-786 (2011).

59. Nogueira, M., Balduccini, M., Gelfond, M., Watson, R. and Barry, M.: A prolog decision support system for the space shuttle. In PADL 2001: Proceedings of the 3rd International Symposium on Practical Aspects of Declarative Languages, Vol. 1990 of LNCS, Springer, pp.169-183. Las Vegas, Nevada, USA (2001).

60. Novas, J.M. and Henning, G.P.: Reactive scheduling framework based on domain knowledge and constraint programming, Computers and Chemical Engineering, 34(12), 2129-2148 (2010).

61. Ouelhadj, D. and Petrovic, S.: A Survey of dynamic scheduling in manufacturing systems, Journal of Scheduling, 12(4), 417-431 (2009).

62. Papadimitriou, C. H.: Games against nature, Journal of Computer and System Sciences, 31(2), 288-301 (1985).

63. Pei, J., Pardalos, P.M., Liu, X., Fan, W., Yang, Sh., Wang, L.: Coordination of production and transportation in supply chain scheduling, Journal of Industrial and Management Optimization, 11(2), 399-419 (2015).

64. Popescu, C, Cavia-Soto, M. and Martinez-Lastra, J. L.: Runtime modeling of flow for dynamic deadlock-free scheduling in service-oriented factory automation systems, IETE Tech Rev, 26(3), 203-212 (2009).

65. Qiao, F., Ma, Y., Li, L. and Yu, H.: A Petri net and extended genetic algorithm combined scheduling method for wafer fabrication, IEEE Transactions on Automation Science and Engineering, 10(1), pp.197-204 (2013).

66. Ramírez, J. and Antonio de, A.: Checking the consistency of a hybrid knowledge base system, Knowledge Based Systems, 20(3), 225-237 (2007).

67. Ricca, F., Grasso, G., Alviano, M., Manna, M. Lio, V. Liritano, S. and Leone, N.: Team-building with answer set programming in the gioia-tauro seaport, Theory and Practice of Logic Programming, 12(3), 361-381 (2012).

68. Rolón, M. and Martínez, E.: Agent-based modeling and simulation of an autonomic manufacturing execution system, Computers in Industry, 63(1), 53-78 (2012).

69. Ruiz, R. and Rodríguez, J.A.V.: The hybrid flow shop scheduling problem, European Journal of Operational Research, 205(1), 1-18 (2010).

70. Sabuncuoglu, I. and Goren, S.: Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research, International Journal of Computer Integrated Manufacturing, 22(2), 138-157 (2009).

71. Samer, M. and Szeider, S.: Constraint satisfaction with bounded treewidth revisited, Journal of Computer and System Sciences, 76(2), pp. 103-114 (2010).

72. Schrader, S., Riggs, W.M. and Smith, R.P.: Choice over uncertainty and ambiguity in technical problem solving, Journal of Engineering and Technology Management, 10(1), 73-99 (1993).

73. Soininen, T. and Niemelä, I.: Developing a declarative rule language for applications in product configuration. In: PADL 1999: Proceedings of the 1st International Workshop on Practical Aspects of Declarative Languages, Vol. 1551 of LNCS, (pp.305 – 319). San Antonio, TX, USA (1999).

74. Son, T.C., Pontelli, E. and Le, T.: Two applications of the ASP-prolog system: decomposable programs and multi-context systems. In PADL 2014: Proceedings of the Sixteenth International Symposium on Practical Aspects of Declarative Languages, (pp.87-103), San Diego, CA., USA (2014).

75. Song, Y., Zhang, M.T., Yi, J., Zhang, L., Zheng, L.: Bottleneck station scheduling in semiconductor assembly and test manufacturing using ant colony optimization, IEEE Transactions on Automation Science and Engineering, 4(4), 569-578 (2007).

76. Tannert, C., Elvers, H. and Jandrig, B.: The ethics of uncertainty: in the light of possible dangers, research becomes a moral duty, EMBO Report, 8(10), 892-896 (2007).

77. Valckenaers, P., Van Brussel, H., Bongaerts, L. and Wyns, J.: Results of the holonic control system benchmark at the KULeuven. In: CIMAT 1994, Proceedings of the Conference Computer Integrated Manufacturing and Automation Technology, Vol. 10-12, (pp.128 - 133).Troy, NY, USA (1994).

78. Van Gelder, A., Ross, A. K. and Schlipf, J. S.: The well-founded semantics for general logic programs, Journal of ACM, 38(3), 620-650 (1991).

79. Vonder, S.V.D., Demeulemeester, E., Herroelen, W.: A classification of predictive-reactive project scheduling procedures, Journal of Scheduling, 10(3), 195-207 (2007).

80. Verfaillie, G. and Jussien, N.: Constraint solving in uncertain and dynamic environments: a survey, Journal Constraints, 10(3), 253-281 (2005).

81. Vieira, G., Herrman, J. and LAntoniou, G.: Verification and correctness issues for nonmonotonic knowledge bases, International Journal of Intelligent Systems, 12(10), 725-738 (1997).

82. Vrba, P., Marík, V. and Kadera, P. MAST: From a toy to real-life manufacturing control. In: SNPD 2012, Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp.428-433. Kyoto, Japan (2012).

83. Wen, H., Fu, L. and Huang, S.: Modeling, scheduling, and prediction in wafer fabrication systems using queueing petri net and genetic algorithm. In: ICRA 2001, IEEE International Conference on Robots and Automation, Vol. 4, (pp.3559 – 3564). Seoul, Korea (2001).

84. Yao, S., Jiang, Z., Li, N., Zhang, H. and Geng, N.A.: Multi-objective dynamic scheduling approach using multiple attribute decision making in semiconductor manufacturing, International Journal of Production Economics, 130(1), 125-133 (2011).

85. Zakaria, A. and Petrovic, S.: Genetic algorithms for match-up rescheduling of the flexible manufacturing systems, Computers and Industrial Engineering, 62(2), 670-686 (2012).

86. Zawadzki, E. P., Platzer, A. and Gordon, G. J.: A generalization of SAT and #SAT for robust policy valuation. In: IJCAI 2013, Proceedings of the International Joint Conference in Artificial Intelligence, pp.2583-2589, Beijing, China (2013).

87. Zeballos, L. J., Castro, P. M. and Méndez, C. A.: Integrated constraint programming scheduling approach for automated wet-etch stations in semiconductor manufacturing, Industrial and Engineering Chemistry Research, 50(3), 1705-1715 (2011).

88. Zhang, L., Madigan, C.F., Moskewicz, M.W. and Malik, S.: Efficient conflict driven learning in boolean satisfiability solver. In: ICCAD 2001, Proceedings of the International Conference of Computer-Aided Design, (pp.279-285). San José, CA, USA (2001).

89. Zhang, D.Z. and Anosike, A.I. (2012). Modeling and simulation of dynamically integrated manufacturing systems, Journal of Intelligent Manufacturing, 23(6), 2367-2382.in, E.: Rescheduling manufacturing systems: a framework of strategies, policies and methods, Journal of Scheduling, 6(1), 39-62 (2003).