Expert Systems with Applications 39 (2012) 10251–10268

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

SmartGantt – An intelligent system for real time rescheduling based on relational reinforcement learning

Jorge Palombarini [a], Ernesto Martínez [b,*]

[a] GISIQ (UTN), Av. Universidad 450, Villa María 5900, Argentina
[b] INGAR (CONICET-UTN), Avellaneda 3657, Santa Fe, S3002 GJC, Argentina

ARTICLE INFO

Keywords:
Manufacturing systems
Real-time rescheduling
Automated planning
Reinforcement learning
Information systems
Relational abstractions

ABSTRACT

With the current trend towards cognitive manufacturing systems to deal with unforeseen events and disturbances that constantly demand real-time repair decisions, learning/reasoning skills and interactive capabilities are important functionalities for rescheduling a shop-floor on the fly taking into account several objectives and goal states. In this work, the automatic generation and update through learning of rescheduling knowledge using simulated transitions of abstract schedule states is proposed. *Deictic* representations of schedules based on focal points are used to define a repair policy which generates a goal-directed sequence of repair operators to face unplanned events and operational disturbances. An industrial example where rescheduling is needed due to the arrival of a new/rush order, or whenever raw material delay/shortage or machine breakdown events occur are discussed using the *SmartGantt* prototype for interactive rescheduling in real-time. *SmartGantt* demonstrates that due date compliance of orders-in-progress, negotiating delivery conditions of new orders and ensuring distributed production control can be dramatically improved by means of relational reinforcement learning and a deictic representation of rescheduling tasks.

1. Introduction

Increasing global competition, a shift from seller markets to buyer markets, mass customization, operational objectives that highlight customer satisfaction and ensuring a highly efficient production, give rise to complex dynamics and on-going disruptive events in industrial environments (Henning & Cerdá, 2000; Zaeh, Reinhart, Ostgathe, Geiger, & Lau, 2010). Moreover, stringent requirements with regard to reactivity, adaptability and traceability in production systems and supply chains are demanded for products, processes and clients all over the product lifecycle. In this context, established production planning and control systems must cope with unplanned events and intrinsic variability in manufacturing environments where difficult-to-predict circumstances occur as soon as plans are released to the shop-floor (Méndez, Cerdá, Harjunkoski, Grossmann, & Fahl, 2006; Vieira, Herrmann, & Lin, 2003). Equipment failures, quality tests demanding reprocessing operations, rush orders, delays in material inputs from previous operations and arrival of new orders give rise to uncertainty in real time schedule execution. In this way, for both human planners and shop floor operators (who interpret the plan) uncertainty in a real manufacturing system is a complex phenomenon that cannot be addressed exclusively through the inclusion of uncertain parameters into problem statement (Aytug, Lawley, McKay, Mohan, & Uzsoy, 2005). Several disturbances and events may produce different impacts depending on the context in which they occur, e.g. operators performance may vary during the week or events arising at night may have a greater impact due to the absence of specialized support personnel, as well as the uncertainty that affects materials availability may disrupt production processes in different ways, depending on product recipes.

The vast majority of the scheduling research does not explicitly consider execution issues such as uncertainty, and implicitly assumes that the global schedule will be executed exactly as it emerges from the algorithm that generates it. The existing body of theory does not address different causes, the context in which uncertainty arises, or the various impacts that might result (McKay & Wiers, 2001; Pinedo, 2005, 2008). Moreover, including additional constraints into global scheduling models significantly increases problem complexity and computational burden, of both the schedule generation and rescheduling tasks, which are (in general) NP-hard (Chieh-Sen, Yi-Chen, & Peng-Jen, 2012). Hence, schedules generated under deterministic assumptions are often suboptimal or even infeasible (Henning, 2009; Li & Ierapetritou, 2008; Vieira et al., 2003; Yagmahan & Yenisey, 2010; Zaeh et al., 2010). As a result, reactive scheduling is heavily dependent on the capability of generating and representing knowledge about strategies for repair-based scheduling in real-time. Finally, producing *satisfactory* schedules rather than optimal ones in reasonable computational

* Corresponding author.
E-mail address: ecmarti@santafe-conicet.gov.ar (E. Martínez).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.176

time, in an integrated manner with enterprise resource planning and manufacturing execution systems is mandatory for responsiveness (Herroelen & Leus, 2004; Trentesaux, 2009; Vieira et al., 2003).

Reactive scheduling literature mainly aims to exploit peculiarities of the specific problem structure (Adhitya, Srinivasan, & Karimi, 2007; Miyashita, 2000; Miyashita & Sycara, 1995; Zhang & Dietterich, 1995; Zhu, Bard, & Yu, 2005; Zweben, Davis, Doun, & Deale, 1993). More recently, Li and Ierapetritou (2008) have incorporated uncertainty in the form of a multi-parametric programming approach for generating rescheduling knowledge for specific events. However, the tricky issue is that resorting to a feature-based representation of schedule state is very inefficient, and generalization to unseen schedule states is highly unreliable (Morales, 2004). Therefore, any learning performed and acquired knowledge are difficult to transfer to unseen scheduling domains, being the user-system interactivity severely affected due to the need of compiling the repair-based strategy for each disruptive event separately. Most of the existing works on rescheduling prioritize schedule efficiency using a mathematical programming approach, in which the repairing logic is not clear to the end-user. In contrast, humans can succeed in rescheduling thousands of tasks and resources by increasingly learning in an interactive way a repair strategy using a natural abstraction of a schedule: a number of objects (tasks and resources) with attributes and relations (precedence, synchronization, etc.) among them. Such conditions, as well as the requirements agility and productivity, together with poor predictability of a shop-floor dynamics, an increasing number of products, reconfigurable manufacturing lines and fluctuations in market conditions demand from production planning and control systems to incorporate higher levels of intelligence.

Today's standard, rigid and hierarchical control architectures in industrial environments have been unable to face with the above challenges, so it is essential to pursue a paradigm shift from off-line planning systems to on-line and closed-loop control systems (Zaeh & Ostgathe, 2009), which take advantage of the ability to act interactively with the user, allowing him to express his preferences in certain points of the decision making process to counteract the effects of unforeseen events, and set different schedule repair goals that prioritize various objectives as such stability, efficiency, or a mix between the two, having into account particular objectives related to customer satisfaction and process efficiency. A promising approach to sustainable improvements in flexibility and adaptability of production systems is the integration of artificial cognitive capabilities, involving perception, reasoning /learning and planning skills (Zaeh et al., 2010). Such ability enables the scheduling system to assess its operation range in an autonomic way, and acquire experience through intensive simulation while performing repair tasks. By integrating learning and planning, the system builds models about the production process, resource and operator capabilities, as well as context information, and at the same time discovers structural patterns and relations using general domain knowledge. Therefore, a scheduling system integrates continuous real-time information from shop-floor sensors/actuators with models that are permanently updated to adapt to a changing environment, and to optimize action selection. At the representation level, it is mandatory to scale up towards a richer language that allows the incorporation of the capabilities mentioned above (Morales, 2003; Van Otterlo, 2009); in that sense, first-order relational representation it's a natural choice because it enables the exploitation of the existence of domain objects and relations (or, properties) over these objects, and make room for *quantification* over objectives (goals), action effects and properties of schedule states (Blockeel, De Raedt, Jacobs, & Demoen, 1999; Džeroski, De Raedt, & Driessens, 2001).

In this work, a novel real-time rescheduling prototype application called *SmartGantt*, which resorts to a relational (deictic) representation of (abstract) schedule states and repair operators with RRL is presented. To learn a near-optimal policy for rescheduling using simulations (Croonenborghs, 2009), an interactive repair-based strategy bearing in mind different goals and scenarios is proposed. To this aim, domain-specific knowledge for reactive scheduling is developed using two general-purpose algorithms already available: TILDE and TG (De Raedt, 2008; Džeroski et al., 2001).

2. Repair-based (re)scheduling in *SmartGantt*

Fig. 1 depicts the repair-based architecture implemented by *SmartGantt*, embedded in a more general setting including an Enterprise Resource Planning System (ERP) and a Manufacturing Execution System with communication and control infrastructures, which integrates artificial cognitive capabilities in resources and processes to include by design flexibility and adaptability in production systems (Trentesaux, 2009). In this approach, search control knowledge about optimal selection of repair operators is generated through reinforcements using a schedule state simulator. In the simulation environment, an instance of the schedule is interactively modified by the learning system which executes control actions using a sequence of repair operators until a repair goal is achieved. In each learning episode, *SmartGantt* receives information from the current schedule situation or state s and then selects a repair operator which is applied to the current schedule, resulting in a new one.

The evaluation of the resulting quality of a schedule after a repair operator has been applied is performed by *SmartGantt* using the simulation environment via an objective or reward function $r(s)$. The learning system then updates its action-value function $Q(s, a)$ that estimates the value or utility of resorting to the chosen repair operator a in a given schedule state s. Such an update is made using a reinforcement learning algorithm (Sutton & Barto, 1998) such as the well-known Q-learning rule, which is showed in Fig. 2. By accumulating enough experiences over many simulated transitions, *SmartGantt* is able to learn an optimal policy for choosing the best repair operator at each schedule state.

The main benefit of applying reinforcement learning techniques such as Q-learning to search control knowledge for improving quality and efficiency of real-time rescheduling is that there is no extra burden on the availability of domain experts, allows online adaptation to a dynamic environment and make room for abstractions that are necessary to deal with large state spaces, e.g. supply chains. For repairing a schedule, *SmartGantt* is given a repair-based *goal* function:

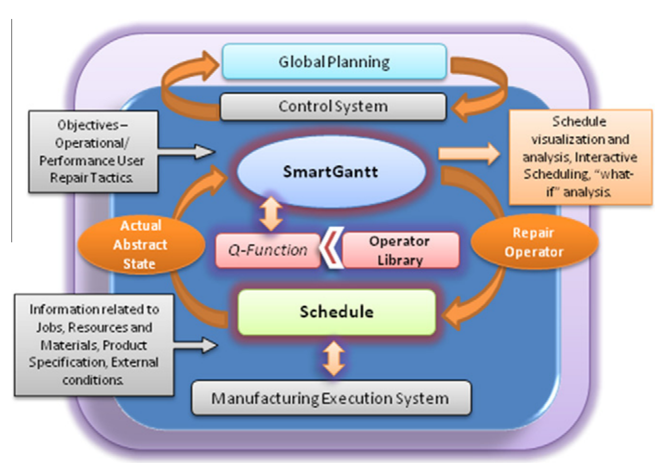

Fig. 1. Repair-based architecture implemented by *SmartGantt*.