International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:6, 2016

Towards Developing a Self-Explanatory Scheduling System Based on a Hybrid Approach

Jian Zheng, Yoshiyasu Takahashi, Yuichi Kobayashi, Tatsuhiro Sato

Abstract**—**In the study, we present a conceptual framework for developing a scheduling system that can generate self-explanatory and easy-understanding schedules. To this end, a user interface is conceived to help planners record factors that are considered crucial in scheduling, as well as internal and external sources relating to such factors. A hybrid approach combining machine learning and constraint programming is developed to generate schedules and the corresponding factors, and accordingly display them on the user interface. Effects of the proposed system on scheduling are discussed, and it is expected that scheduling efficiency and system understandability will be improved, compared with previous scheduling systems.

Keywords**—**Constraint programming, Factors considered in scheduling, machine learning, scheduling system.

I. INTRODUCTION

IMPROVING the efficiency, productivity and service level is a challenge for manufacturing and service industries. To overcome such challenge, planning and scheduling play an important role due to the fact that they directly influence the manufacturing cost, production quality, level of service. During the past decades manufacturing and servicing operations have been increasingly becoming complicated. As a result, planning and scheduling for planners are becoming more challenging. In order to support planning and scheduling, many kinds of systems have been developed [1].

In development of scheduling systems, typically, researchers emphasized on human-system interactions, since automatically generated schedules might be unsatisfactory sometimes. Haynes et al. [2] developed a meeting scheduling system that can take into account user preferences. The system enables users to specify the time of day, day of week, status of other invitees, topic of the meeting. Higgins [3] pointed out that for most scheduling systems using a Gantt chart which is a display of solution is inadequate as the standard interface for human-system interaction in decision making. The author suggested that, for humans to be active partners in decision making, the primary interfaces should display detailed characteristics of jobs in a way that reveals patterns in the data and helps inferential processing. Fagerholt [4] built a decision support system for vessel fleet scheduling, which can easily alter the suggestion from the system through the interface. On the other hand, in order to make scheduling systems more intelligent, some researchers paid attention to big data utilization. Demirkan and Delen [5] proposed a conceptual framework for decision support systems by using big data to leveraging the capacities of the systems. Li [6] utilized historical schedules, internal and external data to create a scheduling system, with the motivation to improve efficiency and productivity of modern manufacturing. In the aforementioned studies, however, it is noted that systems were usually developed as “black boxes”, in a sense that schedules were provided without any explanations. With such systems, users are hard to understand the logic behind the provided schedules. In practice, there are many factors that affect scheduling. For example, when making schedules some planners emphasize on production cost, but others value due date as important. As a result, even with the same order data, schedules made by different planners might be different. Without understanding the logic behind schedules, planners have to spend much time checking and tweaking schedules from the systems based on their own experience. Obviously, this restricts scheduling efficiency and limits transfer of scheduling skills between planners, especially for new planners.

The objective of this study is to develop a self-explanatory scheduling system which can provide not only schedules but also corresponding explanations. To achieve such objective, we first design a user interface that can help planners record crucial factors considered in scheduling, as well as internal and external sources relating to such factors. Second, we develop a hybrid approach to learn planners’ scheduling knowledge recorded by using the user interface, and generate new schedules and corresponding explanations which are accordingly displayed on the user interface. In order to satisfy all constraints imposed by manufacturing machines or customer orders, the hybrid approach is developed to combine machine learning (ML) [7], [8] and constraint programming (CP) [9]. To the authors’ knowledge, this is the first time to present a framework for developing a self-explanatory scheduling system based on the hybrid approach in the literature. It is expected that with the proposed system scheduling will become more understandable and efficient.

The paper is composed of four sections. In the next section, the framework of the system is demonstrated, as well as important components of the system. In Section III, expected effects of the system on scheduling are discussed, and Section IV is devoted to conclusions and further work.

Jian Zheng is with the Hitachi, Ltd., Research & Development Group, Yokohama 244-0817 Japan (phone: +81-50-3135-3434; fax: +81-50-3135-3391; e-mail: jian.zheng.wq@hitachi.com).

Yoshiyasu Takahashi, Yuichi Kobayashi and Tatsuhiro Sato are with the Hitachi, Ltd., Research & Development Group, Yokohama 244-0817 Japan (e-mail: yoshiyasu.takahashi.gq@hitachi.com, yuichi.kobayashi.ma@hitachi.com, tatsuhiro.sato.uq@hitachi.com).

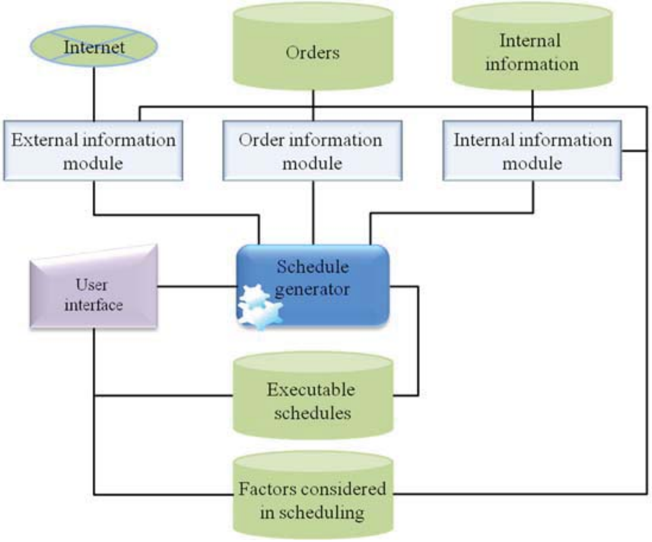

Fig. 1 Framework of the proposed scheduling system

II. FRAMEWORK OF THE SELF-EXPLANATORY SCHEDULING SYSTEM

In this study, the idea behind the development of a self-explanatory system is that we first make field data, such as executable schedules and factors that are considered by planners in scheduling, available. Then, we develop a learning approach and train the learning approach by the field data. When new data are input into the system, the learning approach can generate new schedules, as well as corresponding explanations. To make data available, a user interface is designed to help planners record executable schedules and crucial factors considered in scheduling, as well as external and internal sources relating to such factors. To take constraints imposed by manufacturing machines or customer orders into consideration, the learning approach is developed to combine with CP.

The framework of the proposed system is illustrated in Fig. 1. The system is composed of an external information module, an order information module, an internal information module, a user interface and a schedule generator.

The external information module is used to import external information, such as weather information, from internet. The order information module is utilized to choose orders that should be scheduled in manufacturing or service from the order database. The internal information module is used to import internal information, such as maintenance information and machine operation information, into the system.

The user interface is designed to help planners record the factors that are considered crucial in scheduling, as well as internal and external sources relating to such factors. For example, if a planner considers typhoon information as a crucial factor affecting scheduling, the planner can record such factor and an internet link where typhoon information can be exported. The user interface plays an important role in the development of the system. The structure of the user interface will be introduced in Subsection A.

The schedule generator is used to generate original schedules and corresponding explanations based on order information, external and internal information. The schedule generator is a core component of the system. The approach behind the schedule generator that is used to create schedules and corresponding explanations will be described in Subsection B.

A. User Interface

In the proposed system, the user interface is designed with the following features. First, it can help planners record factors considered in scheduling, as well as external and internal sources relating to such factors. In addition, it can display explanations about original schedules that are generated by the schedule generator, as well as detailed external and internal information.

The user interface used to facilitate system-human interactions in the proposed system is illustrated in Fig. 2. It is composed of 5 information boxes, 4 input buttons, 3 selection buttons.

In the first information box, the original schedule created by the schedule generator is displayed, and factors considered in generating the original schedule are listed in the second box. Important external and internal information are shown in the third and fourth boxes. After the original schedule is checked or modified by a planner, it is treated as an executable schedule and displayed in the executable schedule box.

As we will describe later, several learners are built in the schedule generator. Here, a learner is defined as a ML model whose learning algorithm, input and output can be freely configured. On the user interface, the first selection button is designed to help planners select an appropriate learner, according to their experience. The second selection button is used to facilitate modification of original schedules, such as replacing some orders. The third button is used to help planners select factors that have been recorded in the database.

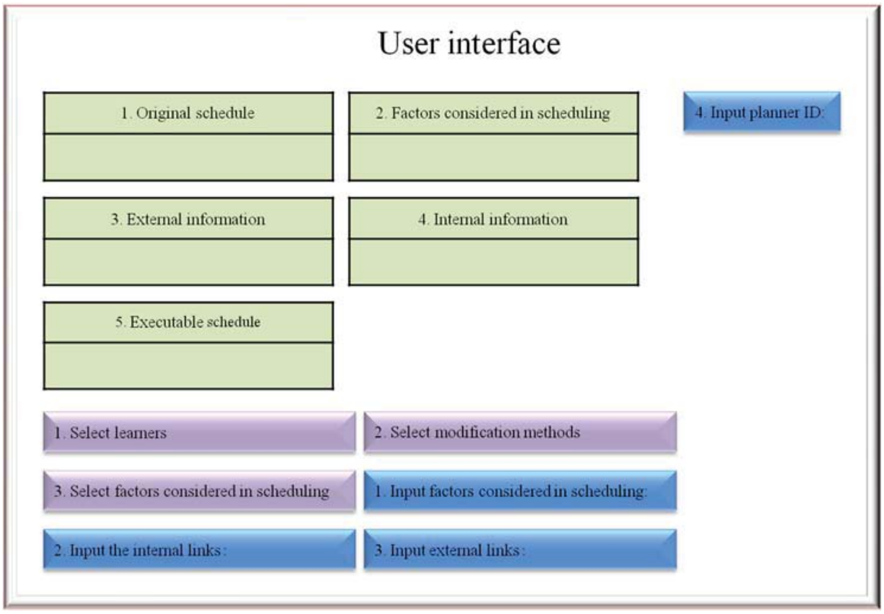

Fig. 2 Structure of the proposed user interface

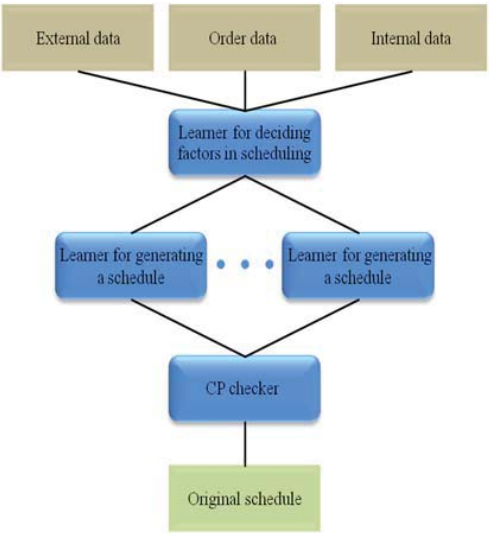

Fig. 3 Structure of the hybrid approach

Among the input buttons, the first button is designed to help planners input the factors that they considered crucial in scheduling. The second and third input buttons are used to assist planners to record internal and external links where external and internal information relating to the crucial factors in scheduling can be exported. For future reference, the fourth button is designed to let planners record their identification numbers.

B. Schedule Generator

The schedule generator which is a core component of the proposed system is composed of a hybrid approach based on nested ML and CP. Fig. 3 shows the structure of the hybrid approach. The approach is composed of 3 levels. The first level contains a learner that is used to decide factors to be considered in scheduling according to external and internal data and order data. The second level contains learners utilized to generate schedules, based on the factors decided in the first level. A CP checker is in the third level, which is developed to check or modify the schedules from the learners in the second level. The schedules checked or modified by the CP checker are treated as original schedules and are output from the schedule generator.

Like a traditional ML approach, learners in the proposed approach should be trained by data. In the next, we show how the hybrid approach works, provided that the learners have been trained.

When external data, internal data and order data are collected and imported into the schedule generator by the external information module, order information module and internal information module, they are configured as inputs of the learner in the first level. Outputs of the learner are factors recorded by planners and represented by zeros or ones. According to the input data, the learner decides which factors should be considered in scheduling. After the factors are determined, they are displayed in the second information box in Fig. 2. Subsequently, in the second level a learner relating to the corresponding factors is used, denoted as the learner N. External data, internal data and order data are set as inputs for the learner N. the output of the learner N is a schedule. Then, the CP checker in the third level checks the schedule in order to make sure that all constraints imposed by manufacturing machines or customer orders are satisfied, and modifies the schedule if necessary. The schedule having been checked or modified is treated as an original schedule and displayed in the first information box on the user interface. If the schedule can’t be modified to satisfy all constraints, the CP checker lists the unsatisfied constraints in the second information box on the user interface, as well.

III. Pre-Evaluation of the System

Because the system is still under development, we just show some expected effects of the system on scheduling in this section. As illustrated in Fig. 4, the proposed system is expected to outperform previous systems in terms of system understandability and system efficiency. More specifically, at the early application stage, due to the lack of available data to learn, there will be no obvious difference in performance of the proposed system and previous systems. However, the user interface of the proposed system which can display factors affecting scheduling or unsatisfied constraints can help planners modify schedules with much ease. Hence, understandability and efficiency of the proposed system will be a little better than that of previous systems.

When sufficient data are available for learning, the nested ML approach adopted in the system will be more powerful and generate more consistent schedules. According to this, improvement in scheduling efficiency of the proposed system will be more obvious, compared with previous systems. Also, with improvement of learning, factors to be considered in scheduling will become more appropriate. Under the support of the proposed user interface, planners will understand which factors have been taken into consideration in scheduling. This will make the final check by planners easy and simple, and facilitate transfer of scheduling skills, especially for new planners. Therefore, in terms of system understandability, the proposed system will also outperform previous systems.

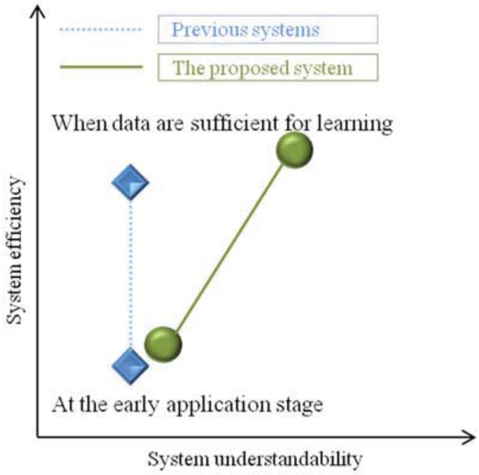

Fig. 4 Comparison results in system efficiency and understandability between the proposed system and previous systems

IV. Conclusions and Further Work

In previous studies, scheduling systems were usually developed as “black boxes”, in a sense that schedules were provided with no explanations. This is not efficient for the final check by planners and limits transfer of scheduling skills between planners. In the study, we discussed a conceptual framework for developing a scheduling system that can generate schedules with easy understanding and self-explanation, based on a hybrid approach.

In the proposed system, a user interface is designed to help planners record crucial factors affecting scheduling, as well as external and internal sources relating to such factors. In addition, a hybrid approach of nested ML and CP is developed to learn planners’ knowledge in deciding crucial factors and planning schedules, and generate new schedules as well as factors considered in scheduling. Accordingly, schedules and corresponding factors generated by the hybrid approach can be displayed on the user interface. It is expected that with the proposed system scheduling will become more understandable and efficient.

Since the system is still under development, some issues, such as incorporation of nested ML and CP, data collection and storage, text and data mining, were not discussed in this study. In our future research, we will show the details regarding such issues, as well as quantitative evaluation results of the system by using field data.

References

[1] Michael Pinedo. *Planning and scheduling in manufacturing and services*, Springer, 2005.

[2] Thomas Haynes, Sandip Sen, Neeraj Arora, and Rajani Nadella, “An automated meeting scheduling system that utilizes user preferences”, *In Proceedings of the first international conference on Autonomous agents*, . ACM, 1997, pp. 308–315

[3] Peter G Higgins, “Interaction in hybrid intelligent scheduling”, *International Journal of Human Factors in Manufacturing*, pp. 185–203. 1996.

[4] Kjetil Fagerholt, “A computer-based decision support system for vessel fleet scheduling - experience and future research”, *Decision Support Systems*, pp. 35–47, 2004.

[5] Haluk Demirkan and Dursun Delen, “Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud”, *Decision Support Systems*, pp. 412–421, 2013.

[6] Xiu Li, Jingdong Song, and Biqing Huang, “A scientific workflow management system architecture and its scheduling based on cloud service platform for manufacturing big data analytics”, *The International Journal of Advanced Manufacturing Technology*, pp. 1–13, 2015.

[7] MG Karlaftis and EI Vlahogianni, “Statistical methods versus neural networks in transportation research: Differences, similarities and some insights”, *Transportation Research Part C: Emerging Technologies,* pp. 387–399, 2011.

[8] MY Rafiq, G Bugmann, and DJ Easterbrook, “Neural network design for engineering applications”, *Computers & Structures*, pp. 1541–1552, 2001.

[9] Francesca Rossi, Peter van Beek, and Walsh Toby, *Handbook of constraint programming*. Elsevier, 2006.