Int. J. Production Economics 145 (2013) 67–77

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

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

Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems

Christoph W. Pickardt [a], Torsten Hildebrandt [b], Jürgen Branke [a,*], Jens Heger [b], Bernd Scholz-Reiter [b]

[a] Warwick Business School, The University of Warwick, Coventry CV4 7AL, UK
[b] BIBA—Bremen Institute of Production and Logistics, Hochschulring 20, 28359 Bremen, Germany

ARTICLE INFO

Article history:
Received 17 February 2012
Accepted 20 October 2012
Available online 30 October 2012

Keywords:
Hyper-heuristics
Dispatching rules
Production scheduling
Semiconductor manufacturing
Evolutionary algorithms
Genetic programming

ABSTRACT

We propose a two-stage hyper-heuristic for the generation of a set of work centre-specific dispatching rules. The approach combines a genetic programming (GP) algorithm that evolves a composite rule from basic job attributes with an evolutionary algorithm (EA) that searches for a good assignment of rules to work centres. The hyper-heuristic is tested against its two components and rules from the literature on a complex dynamic job shop problem from semiconductor manufacturing. Results show that all three hyper-heuristics are able to generate (sets of) rules that achieve a significantly lower mean weighted tardiness than any of the benckmark rules. Moreover, the two-stage approach proves to outperform the GP and EA hyper-heuristic as it optimises on two different heuristic search spaces that appear to tap different optimisation potentials. The resulting rule sets are also robust to most changes in the operating conditions.
© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Production scheduling is concerned with the allocation of resources, e.g. machines, to the processing of a number of jobs. The task is to determine a schedule that optimises a given performance criterion such as the makespan. One of the most complex scheduling problems is the job shop problem, in which each job consists of a number of operations that need to be performed on distinct work centres in a prescribed order, where the order in which a job visits the work centres is job-specific. In this work, a work centre is defined as a set of (1 to m) identical machines with the same functionality.

A widely used approach to real-world scheduling, where problems are often characterised by a highly complex and dynamic environment, are dispatching rules. Dispatching rules are simple heuristics that, whenever a machine is available, determine the job with the highest priority of the jobs waiting to be processed next on that machine. The computation of priorities is typically based on local information, which allows dispatching rules to be executed quickly, irrespective of the complexity of the overall problem. Moreover, because each scheduling decision is made at the latest possible moment, i.e. immediately before its implementation, dispatching rules naturally possess the ability to react to dynamic changes. Other advantages of dispatching rules include their simple and intuitive nature, their ease of implementation within practical settings, and their flexibility to incorporate domain knowledge and expertise (Aytug et al., 2005; Geiger et al., 2006).

On the other hand, the lack of a global perspective on the problem of dispatching rules is also their biggest drawback. They take scheduling decisions on the basis of current local conditions without assessing the negative impact a decision might have on the decision-making at other work centres in the future. The limited horizon of dispatching rules also explains the absence of a single rule that outperforms all others across different shop configurations, operating conditions and objective functions (Blackstone et al., 1982; Haupt, 1989; Holthaus and Rajendran, 1997; Rajendran and Holthaus, 1999). The decision which rule to select generally depends on the specific problem at hand. In addition, some researchers have shown that it can be beneficial to select different rules at different work centres within a shop. This appears to be particularly true for problems where work centres vary with respect to their relative position in the system (LaForge and Barman, 1989; Mahmoodi et al., 1996; Barman, 1997) or their utilisation (Raman et al., 1989; Ruben and Mahmoodi, 1998; Bokhorst et al., 2008), or possess different characteristics altogether (Cigolini et al., 1999; Lee et al., 2003). In summary, the employed rules typically have to be customised to the problem in order to tap the full potential of a dispatching rule-based approach.

The development of customised dispatching rules is usually a tedious procedure requiring a significant amount of expertise, coding-effort and time. The challenge is to design local, decentralised rules which result in a good global performance of a complex production environment. Generally, this is achieved by a trial-and-error procedure, with candidate rules tested in a simulation model of the considered manufacturing system, modified,

* Corresponding author.
E-mail addresses: christoph.pickardt@warwick.ac.uk (C.W. Pickardt), hil@biba.uni-bremen.de (T. Hildebrandt), juergen.branke@wbs.ac.uk (J. Branke), heg@biba.uni-bremen.de (J. Heger), bsr@biba.uni-bremen.de (B. Scholz-Reiter).

0925-5273/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2012.10.016

and retested until they fulfill the requirements for actual implementation (Geiger et al., 2006). This process can be automated by a hyper-heuristic. Hyper-heuristics are optimisation methods that operate on a search space of heuristics (Burke et al., 2010). In this work, evolutionary algorithms (EAs) are employed as hyper-heuristics to search for effective dispatching rules, and discrete-event simulation is used to evaluate the evolved rules.

In a previous paper (Pickardt et al., 2010), we apply a hyper-heuristic that is based on a special type of EA called genetic programming (GP) to create a single dispatching rule for a complex and dynamic job shop from semiconductor manufacturing. Here, we extend the method with another EA that, in a second stage, assigns a different dispatching rule to each work centre in the shop. This two-stage hyper-heuristic is tested by comparing its performance to that of the original GP hyper-heuristic and the standard rule-assignment hyper-heuristic without access to evolved rules.

The paper is organised as follows. Section 2 reviews the related literature, followed by a presentation of the three hyper-heuristics in Section 3. These are applied to a scenario from semiconductor manufacturing, described in Section 4 and results are reported in Section 5. Some investigations of the robustness of the generated dispatching rule sets are done in Section 6, and the paper concludes with a summary and some suggestions for future work.

2. Literature review

An early paper related to the generation of composite dispatching rules whose priority indices are mathematical functions of several job attributes is the one by Hershauer and Ebert (1975). They define composite rules as the weighted sum or product of common priority indices, and use Hooke–Jeeves pattern search to find the best weights for a job shop problem. They find that the effectiveness of their method strongly depends on aspects such as the format of the composite rules and the starting solution of the search. In face of this, GP seems very suitable as it is flexible to create rules of different formats and lengths, and like any EA operates on a set of (starting) solutions. Atlan et al. (1994) employ GP to compose a dispatching rule for a particular job shop instance. The resulting rules obtain (near-)optimal solutions and are robust with respect to perturbations in processing times.

Several recent studies have used GP as a hyper-heuristic to learn a new composite rule that outperforms manually developed benchmark rules on a class of problems. Dimopoulos and Zalzala (2001) and Jakobović and Budin (2006) address various single machine problems with due date-related measures, and report that the rules evolved by GP in most cases outperform the rules from the literature. Geiger et al. (2006) apply a GP hyper-heuristic to a range of single machine problems, which finds optimal dispatching rules where they are known and yields competitive rules for all other problems. In a follow-up paper, Geiger and Uzsoy (2008) use their hyper-heuristic to evolve dispatching rules for a batch processing machine that can process several jobs together in a batch, and again manage to generate good rules that are optimal in some cases. More complicated problem environments are considered by Jakobović et al. (2007), who use GP to create rules for several parallel machine problems with and without sequence-dependant setup times and Tay and Ho (2008), who address a flexible job shop problem, where each work centre contains several machines in parallel. Both studies report the approach to use a GP hyper-heuristic for the generation of composite rules to be successful. Interestingly, the dominance of automatically generated rules seems to become more pronounced for more complex problems (Jakobović et al., 2007), supporting the intuition that the potential of hyper-heuristics is highest when it is difficult to design effective rules manually. Some researchers have proposed alternative methods for the automatic creation of dispatching rules. Olafsson and Li (2010) combine an EA with a decision tree algorithm to learn new rules. Nie et al. (2010) propagate the use of gene expression programming, which is based on similar ideas as GP, for this purpose. Both papers address only single machine problems.

GP-based rule generation has also been successfully used to improve scheduling algorithms of which dispatching rules form an integral part. Yin et al. (2003) apply this technique to evolve predictive scheduling heuristics which produce schedules that are robust to unpredictable breakdowns of machines. In a similar fashion, Vázquez-Rodríguez and Ochoa (2011) create effective variants of the NEH heuristic for different permutation flow shop problems by modifying the dispatching rule underlying the heuristic.

The above publications indicate that GP hyper-heuristics for the generation of dispatching rules is a promising approach. However, they predominantly investigate relatively simple and static problems and allow rules to access future information that is typically unavailable, such as the number of jobs still to arrive or their release dates. Since dispatching rules are more likely to be employed where they are most beneficial, namely in complex, dynamically changing environments, we have previously applied the approach to such problem environments with promising results (Hildebrandt et al., 2010; Pickardt et al., 2010).

The optimisation problem of assigning each work centre the best of a set of given rules so that the rule combination results in a good shop performance has been addressed in various ways. Pierreval and Ralambondrainy (1990), Pierreval (1992) and El-Bouri and Shah (2006) use machine learning techniques, in particular neural networks, that base the choice of an appropriate rule on properties such as the workload distribution or relative position of a work centre. Ishii and Talavage (1994) design a heuristic search algorithm that, starting from a base rule, sequentially selects the best of a given set of dispatching rules for each individual work centre, where work centres are considered roughly in decreasing order of their utilisation. They apply the algorithm to various job shop problems and find combinations of rules that dominate the individual rules. A drawback of this procedure is that it is not always easy to identify the extent to which work centres are critical, especially in the presence of sequence-dependant setups and batch processing. Yang et al. (2007) develop an EA hyper-heuristic that selects different rules for different work centres. They apply the hyper-heuristic to the problem of minimising mean tardiness in a flexible flow shop, for which it discovers rule combinations that clearly outperform some benchmark rules.

A natural extension to the studies above is the design of hyper-heuristics that exploit the two potentials of composite rules and sets of work centre-specific rules at the same time. Baek et al. (1998) propose a procedure that sequentially generates a composite rule for each work centre one-by-one. They apply it to two flexible flow shops, for which it is able to learn rules that yield a significantly lower mean flow time than all benchmark rules. In a follow up paper, Baek and Yoon (2002) develop a coevolutionary algorithm (CoEA) that evolves the work-centre specific composite rules in parallel. Applied to a flexible job shop problem with the mean tardiness objective, they find the CoEA hyper-heuristic to be more effective and efficient than the sequential procedure suggested in their earlier paper. Geiger et al. (2006) use GP to generate one composite rule for each machine to address the two machine-flow shop makespan problem. They report that the evolved rule set resembles the behaviour of the optimal Johnson algorithm.

The main problem with the development of hyper-heuristics for the automatic generation of entire sets of work centre-specific composite rules is the size of the search space, which grows exponentially both in the number of components that define a

Download English Version:

https://daneshyari.com/en/article/5080496

Download Persian Version:

https://daneshyari.com/article/5080496

Daneshyari.com