Computers & Operations Research 66 (2016) 215–224

Contents lists available at ScienceDirect

Computers & Operations Research

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

A hybrid differential evolution approach based on surrogate modelling for scheduling bottleneck stages

Jing-hua Hao, Min Liu*, Jian-hua Lin, Cheng Wu

Department of Automation, Tsinghua National laboratory for information science and technology, Tsinghua University, Beijing, China

ARTICLE INFO

Available online 28 August 2015

Keywords:
Scheduling
Surrogate model
Differential evolution
Extreme learning machine

ABSTRACT

Surrogate modelling based optimization has attracted much attention due to its ability of solving expensive-to-evaluate optimization problems, and a large majority of successful applications from various fields have been reported in literature. However, little effort has been devoted to solve scheduling problems through surrogate modelling, since evaluation for a given complete schedule of these complex problems is computationally cheap in most cases. In this paper, we develop a hybrid approach for solving the bottleneck stage scheduling problem (BSP) using the surrogate modelling technique. In our approach, we firstly transform the original problem into an expensive-to-evaluate optimization problem by cutting the original schedule into two partial schedules using decomposition, then a surrogate model is introduced to, quickly but crudely, evaluate a given partial schedule. Based on the surrogate model, we propose a differential evolution (DE) algorithm for solving BSPs in which a novel mechanism is developed to efficiently utilize the advantage of the surrogate model to enhance the performance of DE. Also, an improved adaptive proximity-based method is introduced to balance the exploration and exploitation during the evolutionary process of DE. Considering that data for training the surrogate model is generated at different iteration of DE, we adopt an incremental extreme learning machine as the surrogate model to reduce the computational cost while preserving good generalization performance. Extensive computational experiments demonstrate that significant improvements have been obtained by the proposed surrogate-modelling based approach.

1. Background and motivation

Bottleneck stage exists in almost all of the manufacturing facilities such as the weaving stage in textile industry, the assembly stage in mechanical industry, the photoetching stage and the diffusion stage in wafer fabrication industry. Effective scheduling on bottleneck stages can significantly improve the performance of the overall production system, as bottleneck stages are generally most influential to factory-level KPIs according to the principle of *wooden barrel*. This paper studies bottleneck stage scheduling problems (BSPs) in manufacturing facilities, where multiple product types are produced with the objective of minimizing the total holding cost, i.e., the cost of total weighted flowtime of jobs. Schedulers in facilities are required at different time to make out a complete schedule in which jobs are allocated to machines and sequenced for each machine. The above problems are often modelled in literature as parallel machine scheduling problems with release times and complicated constraints [1–5].

There are a large majority of papers in literature concerning the topic of modelling and solving bottleneck stage scheduling problems, where different methods are extensively proposed including dispatching rules, exact methods, and meta-heuristics. Monch et al. [3] give a comprehensive survey on models, solution techniques and future challenges of scheduling problems in semiconductor industry, where they point out that there are several important bottleneck stages in the wafer fabrication process including photoetching, ion implantation, etching, diffusion, and oxidation. Chien and Chen [4] develop a novel genetic algorithm for scheduling furnace machines in diffusion area while considering time constraints and the makespan objective. Wang et al. [5] present a hybrid dynamic harmony search algorithm for solving identical machine scheduling problem, which can be used in scheduling bottleneck stages. Lamothe et al. [6] propose some effective dispatching rules for scheduling bottleneck stages with the existence of setup times and calendar constraints. Ranjbar et al. [7] present two branch and bound algorithms for parallel machine scheduling problems with the objective of finding a robust schedule under stochastic environments.

* Corresponding author.
E-mail addresses: haojinghua@tsinghua.edu.cn (J.-h. Hao), lium@tsinghua.edu.cn (M. Liu), ljh10@tsinghua.edu.cn (J.-h. Lin), wuc@tsinghua.edu.cn (C. Wu).

http://dx.doi.org/10.1016/j.cor.2015.08.005
0305-0548/

However, due to the NP-Hard nature of most BSPs, although some successful applications are reported in literature, existing methods are not very powerful for handling real-world large-scale instances [3]. The key reason is that the permitted running time for scheduling algorithms is often seriously restricted in real-world applications, especially under online rescheduling environments. For example, in semiconductor industry, schedulers are often required to make out a schedule for tens of machines within several minutes. Therefore, developing algorithms that can obtain good solutions in limited time turns to be very important and helpful to schedulers. Dispatching rules are often used in real-time scheduling, but they are myopic as most of them only incorporates local information to give a scheduling decision. Exact methods, such as dynamic programming and branch and bound algorithms, are often encountered with curse of dimensionality and are computationally intractable when handling large-scale instances. In the last two decades, with the development of more and more powerful computers, population-based meta-heuristics such as genetic algorithm, particle swarm optimization and differential evolution, have gained much attention and numerous applications for various optimization problems are proposed. However, when handling large-scale problems, they still need much computation time in order to obtain a good solution due to a large number of simulations in the evolutionary process.

Surrogate modelling [8] is a powerful tool for solving expensive-to-evaluate optimization problems, and has gained much attention from different disciplines recently. The inherent idea is that for expensive-to-evaluate systems, a surrogate model is introduced to approximate the original system (e.g., simulation system, semi-physical system and physical system) and therefore to provide valuable information without actually running the original expensive-to-evaluate system. The surrogate model is applicable in two cases: (1) it is very expensive, or even infeasible due to safety considerations, to run the physical or semi-physical system with arbitrary inputs, (2) it is very time-consuming to run the simulation system based on the large-scale and complex simulation model. In these two cases, surrogate modelling is used as an effective tool to assist designers in finding better configuration of the system. Successful applications on problems from various fields, including fast optimization of energy system [9], aircraft engineering design [10] and aerospace structure optimization [11] have been reported in literature.

Motivated by successful applications of surrogate modelling technique, in this paper we try to incorporate this technique for solving BSPs mentioned above, which is a typical NP-Hard combinatorial optimization problem. However, evaluation for a single schedule of BSP is computationally cheap in general sense (according to our experiments, evaluation of a complete schedule of a 1000×50 BSP instance costs less than 0.5 s on a 1.7 GHz Processor and 4G RAM, where the number of jobs is 1000 and the number of machines is 50), therefore, little effort in literature has been devoted to solve scheduling problems by surrogate modelling approach. In this paper, we develop a novel approach to transform the BSP into an expensive-to-evaluate problem by cutting the original schedule S into two partial schedules S_1, S_2 using machine-based decomposition. The new problem after decomposition is expensive-to-evaluate in the sense that when evaluating a given partial schedule S_1, we need to introduce a generally time-consuming algorithm to search a good-enough schedule in the space of S_2 on the condition that S_1 is determined.

Under the above transformation, in our solution framework a surrogate model is introduced to evaluate, quickly but not necessarily accurately, the given partial schedule through the corresponding feature attributes. We select the extreme learning machine (ELM) as the basis of the surrogate model due to its high efficiency of training and good generalization performance on regression problems. Also, the hidden layer of ELM needs not to be tuned in the training process, while in neural networks, tuning parameters of hidden layer is usually a difficult work. Furthermore, considering that training data for the surrogate model is generated at different iterations of the algorithm, we use the incremental form of ELM (IELM) in order to accelerate the training procedure while preserving good generalization performance. Our previous work [12] also shows the superiority of this methodology.

Based on the IELM-based surrogate model, we present a differential evolution (DE) algorithm for solving BSPs in which a novel mechanism is developed to efficiently utilize the advantage of the surrogate model to enhance the performance of the algorithm. Also, an improved adaptive proximity-based method is introduced to balance the exploration and exploitation during the evolutionary process of DE.

This paper is organized as follows. Section 2 gives the detailed description of the BSP problem, and formulates the problem as a mixed integer programming model. In Section 3, the decomposition of the problem is illustrated. Section 4 describes the surrogate modelling based differential evolution algorithm. Simulation results and conclusions are given in Sections 5 and 6, respectively.

2. Problem description and formulation

Scheduling of bottleneck stages is often modelled as parallel machine scheduling problem with different constraints and objectives in literature. Here we describe and formulate the studied BSP problem as below.

There are totally n independent jobs $N = \{1, 2, \ldots, n\}$, m parallel machines $M = \{1, 2, \ldots, m\}$. Jobs are required to be processed on machines to minimize some performance criterions. Each job $j \in N$ is to be processed by exactly one of the machines, and has a processing time p_j, a weight w_j, and an estimated arriving time r_j. Also, there are eligibility constraints between machines and jobs, i.e., each job corresponds to a set of machines μ_j to which the job can be assigned due to technological considerations. If a machine is ready for processing a job but the job has not arrived yet, it stays idle until the job arrives. The problem is to assign each job to a machine and sequence jobs for each machine such that the total weighted flow time is minimized.

Assume $am_j (am_j \in M)$ to be the assigned machine for job j, e.g. $am_j = 2$ represents job j is to be processed on machine 2#, st_j be the starting time of job j, then the above problem can be formulated as a mixed integer linear programming model as follows.

$$\min TFT = \sum_{j=1}^{n} w_j f_j = \sum_{j=1}^{n} w_j \left(st_j + p_j - r_j \right) \quad (1)$$

s.t.

$$st_j \geq r_j, \quad j = 1, 2, \cdots, n \quad (2)$$

$$am_j \in \mu_j, \quad j = 1, 2, \cdots, n \quad (3)$$

$$st_{k,j} - st_{k,i} \geq p_i \cup st_{k,i} - st_{k,j} \geq p_j$$

$$k = 1, 2, \cdots, m, \quad i, j = 1, 2, \cdots, n, \quad i \neq j \quad (4)$$

Eq. (1) represents the objective of the considered problem, Eq. (2) means the starting time of each job should be arranged after the job arrives, Eq. (3) means the allocation of jobs should satisfy machine eligibility constraints, and Eq. (4) restricts that a machine can process at most a job at a time. The above problem can be denoted as $P|\mu_j|\sum w_j f_j$ using the 3-field notation of Graham [13].

Such a mathematical formulation covers many industrial bottleneck stage scheduling problems such as the weaving stage in