Computers & Operations Research 44 (2014) 115–136

Contents lists available at ScienceDirect

Computers & Operations Research

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

A multi-agent system for the weighted earliness tardiness parallel machine problem

S. Polyakovskiy [a], R. M'Hallah [b],*

[a] School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia
[b] Department of Statistics and Operations Research, College of Science, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait

ARTICLE INFO

Keywords:
Parallel machine scheduling
Multi-agent systems
Artificial intelligence
Earliness
Tardiness
Local search

ABSTRACT

This paper studies the weighted earliness tardiness parallel machine problem where jobs have different processing times and distinct due dates. This NP hard problem arises in most just-in-time production environments. It is herein modeled as a mixed integer program, and solved using MAS^H, a deterministic heuristic based on multi-agent systems. MAS^H has three types of agents: I, G, and M. The I-agents are free jobs that need to be scheduled, whereas the G-agents are groups of jobs already assigned to machines. The M-agent acts as the system's manager of the independent intelligent I- and G-agents, which are driven by their own goals, fitness assessments, and context-dependent decision rules. The I- and G-agents employ exact and approximate approaches as part of their decisional process while the M-agent uses local search mechanisms to improve their (partial) solutions. The design of MAS^H is innovative in the way its intelligent agents determine bottleneck clusters and resolve conflicts for time slots. The numerical results provide computational evidence of the efficiency of MAS^H, whose performance on benchmark instances from the literature is superior to that of existing approaches. The success of MAS^H and its modularity make it a viable alternative to more complex manufacturing problems. Most importantly, they demonstrate the benefits of the hybridization of artificial intelligence and operations research.

1. Introduction

Minimizing total earliness and tardiness penalties gained importance with the emergence of Just-In-Time. Companies adhering to this philosophy strive to complete jobs exactly on their due dates. A job that completes before its due date incurs an earliness penalty due to the holding cost whereas a job that completes after its due date generates a tardiness cost due to late charges, express delivery, and lost sales. In such environments, classical measures of performance (e.g., makespan, mean tardiness, mean lateness) are not suitable.

This paper studies the deterministic non-identical parallel machine weighted earliness tardiness problem, $R|d_j|\sum \alpha_j E_j + \beta_j T_j$, denoted hereafter PWET. Every job of a set of n independent jobs, available at time zero, is to be scheduled, without preemption, on one of m non-identical parallel machines. A job j, $j=1,\dots,n$, is characterized by a known distinct due date d_j, a processing time p_{ij} on machine i, $i=1,\dots,m$, a per unit cost of earliness α_j, and a per unit cost of tardiness β_j. A machine can process only one job at a time, and can be idle. The objective is to find a schedule that minimizes the sum of weighted earliness and tardiness (WET) of the n jobs.

Herein, PWET is approximately solved using MAS^H, a heuristic based on a multi-agent system (MAS). The superscript H in MAS^H distinguishes between the general MAS framework and the proposed heuristic. Four factors motivate this choice of approach. First, MAS is in general best suited for modular applications [25], and PWET is modular. Its modules are the jobs and machines. Second, MAS is particularly useful when the problem has no alternative optimization technique or when existing ones are highly expensive, not efficient, fast, or easily applicable [16]. This is the case of PWET where large-sized instances can't be solved exactly, and existing approximate approaches for variants of the problem are not fast and efficient. Third, MAS^H adopts a decentralized decisional process in the sense that some decisions are delegated to low-level agents in lieu of emanating from the higher-level system's manager [6]. Differently stated, there is no single intelligent entity which provides a monolithic solution to the scheduling problem. This characteristic allows a higher level of autonomy for lower-level agents in their decision making process. It is applied to enhance the system's performance. On the other hand, existing approximate optimization techniques that tackled variations of PWET are centralized in the sense that the optimization technique decides the starting time of every job and its

---

* Corresponding author. Tel.: +965 669 14150; fax: +965 248 37332.
E-mail addresses: maxles@yandex.ru (S. Polyakovskiy), rym.mhallah@ku.edu.kw, rymmha@yahoo.com (R. M'Hallah).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.10.013

machine assignment. Fourth and last, PWET involves tradeoffs and bargaining among jobs and competition for resources. These activities cannot be correctly described by a centralized approach, while they can be captured by MAS^H. (cf. [4] for a comparison of agent-based versus classical optimization techniques.)

The efficient adaptation of a search heuristic to any problem requires a thorough preliminary study of the problem so that its specificities are exploited during the design stage of the heuristic. Like genetic algorithms and ant colonies, MASs constitute a search framework which might not yield high performance heuristics if these latter are not dotted with the appropriate control parameters, decoding mechanisms, etc. Therefore, the successful implementation of the proposed MAS^H depends upon the definition of the agents, the elaboration of their competition and cooperation mechanisms, and the construction of their decisional processes.

MAS^H represents a new design of multi-agent based heuristics. It employs an M-agent that acts as the system's manager of two types of low-level rational agents: free jobs that need to be scheduled, and groups of jobs already assigned to machines. MAS^H bases the agents' decisional processes on “if–then” decision rules that are supported by exact and approximate optimization techniques. It federates these decisions via a competition and negotiation model that is partially inspired from [26]. Thus, its solution construction concept is different from existing approaches in the literature [1,8,21]; in particular, in the way its agents determine bottleneck clusters and resolve conflicts among jobs competing for the same time slot. Moreover, it has an architecture that speeds the required computations. Most importantly, MAS^H demonstrates the potential of the hybridization of artificial intelligence and operations research. Embedding exact optimization techniques into the agents' decisional processes and subjecting a (partial) solution to a local search are, respectively, low and high-level hybridizations whose success highlights the “complementary characteristics of MAS and classical heuristics” [4]. In addition to its innovative aspects and scientific merit, MAS^H can solve PWET when it arises as a subproblem in complex industrial settings; for example, for scheduling jobs on parallel cutters for on-line furniture manufacturing [27].

This paper details the proposed MAS^H and tests its performance on two sets of benchmark instances. The results demonstrate the competitiveness of MAS^H in terms of better local optima and reduced run times; thus providing computational proof of the successful implementation of MAS^H. Section 2 surveys the literature on PWET. Section 3 models PWET as a mixed integer program. Section 4 details MAS^H. Section 5 presents the results. Finally, Section 6 is a summary.

2. Literature review

Most of the literature on parallel machine earliness tardiness problems determines a common due date and subsequently schedules the jobs [14,20]. Alidaee and Panwalkar [2] and Kubiak et al. [19] design an $O(n^3)$ algorithm for $R|d_j=d|\sum E_j+T_j$. De et al. [9] show that $P|d_j=d_{unres}|\sum w_j(E_j+T_j)$ is NP-hard for $m=2$ and strongly NP-hard for $m>2$, where the non-restrictive common due date d_{unres} is large enough not to constrain the scheduling process. They apply a pseudo polynomial dynamic program to problems with 30 jobs but whose characteristics are small integers. Federgruen and Mosheiov [11] derive a lower bound and a heuristic for $P|d_j=d_{unres}|\sum w_j(E_j+T_j)$. Chen and Powell [7] model $P|d_j=d_{unres}|\sum w_j(E_j+T_j)$ as an integer program, and decompose it into a set partitioning formulation with side constraints. Biskup and Cheng [5] show that $P|d_j=d, p_j=p|\sum \alpha E_j+\beta T_j+\delta d$ is polynomially solvable, and present a heuristic for the case with distinct processing times. Sun and Wang [30] show that $P|d_j=\ d, w_j=p_j|\sum w_j(E_j+T_j)$ is NP-hard, and solve it via dynamic programming. Solis and Sourd [29] apply a local search to $P|d_j=d|\sum \alpha_j E_j+\beta_j T_j$. Drobouchevitch and Sidney [10] consider $Q|p_j=p, d_j=d|f(E_j, T_j, d)$ where the decision variable d minimizes a function of earliness, tardiness and due date penalties. Mosheiov and Sarig [24] develop a constant-time algorithm for the case with two uniform machines. Gerstl and Mosheiov [13] study the uniform parallel machine problem where jobs belong to one of k classes and the jobs of a class share a common due-date. They minimize the maximum earliness/tardiness cost and the sum over all k classes of the maximum earliness/tardiness cost of the jobs of each class. Bank and Werner [3] develop a heuristic for the unrelated parallel machine problem with a common due date, release dates, and linear earliness and tardiness penalties.

In contrast to the previous research, Mason et al. [23], Kedad-Sidhoum et al. [18], and M'Hallah and Al-Khamis [22] consider the due dates distinct and known. They focus on the identical machine case. The formers study the unweighed case (i.e. $\alpha_j=\beta_j=1$), whereas the latter considers the general weighed case with the earliness and tardiness penalties not necessarily equal. Mason et al. [23] apply a moving block heuristic and present a mixed integer program based lower bound. Kedad-Sidhoum et al. [18] propose two lower bounds and assess their tightness. M'Hallah and Alkhamis [22] model their problem as a mixed-integer program which exactly solves small-sized instances. For large instances, they design hybrid heuristics which make steepest descent and simulated annealing collaborate with genetic algorithms via different levels and types of hybridization. Variants of PWET are available. For example, Toksari and Güne [31] consider the effects of learning and deterioration. Polyakovskiy and M'Hallah [27] tackle a real life problem that occurs in the furniture industry. Orders are received online. The furniture plant has to decide how to assemble the components of the orders into groups and how to cut them from two-dimensional boards. The objectives of the plant are to maximize the utilization of the boards and minimize the weighted earliness tardiness of the components of the orders as to avoid their temporary storage and blocking the next stations. Thus, PWET was a subproblem to the multiple objective online scheduling problem. It was tackled via a simple heuristic.

This paper addresses $R|d_j|\sum \alpha_j E_j+\beta_j T_j$. It proposes a multi-agent deterministic approach which explores the specificities of the problem to search for a near optimum in lieu of relying on mere enumeration. This heuristic defines its agents in a unique way that distinguishes it from existing MAS implementations for scheduling problems. It is faster and obtains better local minima than the hybrid search H4 of [22], and the search method, denoted hereafter KSS, of [18] for the identical machine case.

3. A mathematical model

There are few mathematical programming models for PWET. Herein, the precedence-based mixed integer linear programming formulation is extended to the non-identical machine case. It uses six types of decision variables. Let x_{ikj}, $i=1,\ldots,m$, $k=1,\ldots,n$, $j=1,\ldots,n$, $k\neq j$, equal 1 if job k *immediately precedes* job j and *both k and j are assigned to the same machine* i, and 0 otherwise. In addition, let ε_{ij}, $i=1,\ldots,m$, $j=1,\ldots,n$, equal 1, if job j is the first job on machine i, and 0 otherwise. Let S_j, $j=1,\ldots,n$, and C_j, $j=1,\ldots,n$, denote the starting time and completion time of job j, respectively. When j is processed on machine $i\in\{1,\ldots,m\}$, $C_j=S_j+p_{ij}$. Finally, let $E_j=\max\{0, d_j-C_j\}$ and $T_j=\max\{0, C_j-d_j\}$ designate the earliness and tardiness of job j, $j=1,\ldots,n$. The first two decision variables are binary whereas