Computers and Operations Research 90 (2018) 110–124

Contents lists available at ScienceDirect

Computers and Operations Research

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

Total completion time minimization for machine scheduling problem under time windows constraints with jobs' linear processing rate function

Nhan-Quy Nguyen[a,*], Farouk Yalaoui[a], Lionel Amodeo[a], Hicham Chehade[a], Pascal Toggenburger[b]

[a] ICD, LOSI, Universite de Technologie de Troyes, UMR 6281, CNRS, France
[b] Park'nPlug, 1-3-5 rue de l'Abbe Roger Derry - 75015 Paris, France

ARTICLE INFO

Article history:
Received 2 March 2016
Revised 31 May 2017
Accepted 14 September 2017
Available online 18 September 2017

Keywords:
Total completion time
Strip packing
Resource allocation
Linear processing rate function

ABSTRACT

In this paper, we consider an identical parallel machine scheduling problem with a single additional resource. The processing rate of a job is defined by a linear resource consumption function. The addressed problem takes into consideration two new constraints. The first is the time-varying total available resource. The second new constraint limits the resource consumption incrementation of each job on two consecutive periods of time. Moreover, jobs have bounded resource consumption, arrival times and deadlines. Many practical applications, such as the electrical charging scheduling, can find interests in our works. Our contributions are two-folds. First, we introduce a Mixed-Integer-Linear-Program (MILP) to formulate the problem. Then, we present a heuristic consisting of two phases: a feasible solution construction phase using geometrical strip packing and a solution improvement phase. The heuristic is proven to be very efficiency for dealing with the problem throughout the numerical experiments.

1. Introduction

In classical machine scheduling problems, processing times of jobs are known in advance, which are assumed to be constant. However, in practical situations, the processing rate of a job can be controllable by the amount of resource allocated to it, causing the variation of processing times. The problem raised in the context of the allocation of resource driving commonly the treatment of different jobs to be scheduled such as workforce, hydraulic/pneumatic power source, financial investment... The bigger amount resource a job receives at a time, the faster it can reach the final state. For example, the fueling of fleets of boat problems (Dror et al., 1987): the refueling is controlled by pumps driven by a common power source. At a time, the power allocation decides the pumping capacity, so it controls the flow of fuel refilled into each boat tank. Janiak (1991) described another classical practical application of the problem: the ingot preheating process in steel mills problem, which is a trade-off between the gas flow intensity and the preheating and rolling time of steel ingots. Recently, in the management of the scalable and massively parallel computing, the number of processors can be thousands, so this latter can be considered as a common continuous resource required for tasks to be computed (Jozefowska and Weglarz, 1998). Introduced in 1970's by Weglarz (1976) by the term of discrete-continuous scheduling problem (DCSP), the problem can be described as an identical parallel machine scheduling with additional continuously divisible and renewable resources. The processing rate of a job at a time is defined by a non-decreasing function called *processing rate function*.

As a particular class of scheduling problems with controllable processing times whose classifications is mentioned in the studied surveys (Shabtay and Kaspi, 2006; Shabtay and Steiner, 2007; Tseng et al., 2009; Weglarz et al., 2011), DCSP has been studied mainly in two classes of problems, known as the trade-off of resource consumption and time. The first class is the minimization of the resource consumption subject to a given upper bound of makespan. Gorczyca and Janiak (2010) proposed an exact algorithm and approximation algorithms solving the particular problem called $PU_{\max}$ which tends to minimize the resource level, i.e., the maximum resource consumption at each time instance. The second class is the minimization of scheduling criterion (makespan, total completion time, total tardiness...) (Akturk et al., 2007; Shabtay and Kaspi, 2004; Xu et al., 2010) subject to an upper bound of renewable resources. In the majority of published works, this particular class of problem is considered, in which jobs

* Corresponding author.
E-mail addresses: nhan_quy.nguyen@utt.fr, nhanquy.ngn@gmail.com (N.-Q. Nguyen).

http://dx.doi.org/10.1016/j.cor.2017.09.015
0305-0548/

are supposed to be non-preemptable and are all available at the start of scheduling, having the same deadline at the end of the scheduling horizon. In addition, the total resource at every time is presumed as a constant value, and the objective is to minimize scheduling length. Jozefowska and Weglarz (1998) divided the general problem into two subproblems. The first one is to find a sequence of jobs to be assigned to machines (in the case where the number of machines is less than the number of jobs) and the second one is to find a continuous resource that optimizes the scheduling performance measure. Jozefowska and Weglarz proved that with a given possible sequence of jobs, a semi-optimal solution could be found analytically if the processing rate function is not greater than linear and is concave, having the form $f_i = h_i u_i^{\frac{1}{a}}$. i is the job index, u_i is the amount of resource allotted to job, $a \in \{1, 2, 3, 4\}$ and h_i is a linear coefficient. A potentially optimal set (POS) is defined as a set of the feasible sequence which contains at least one sequence corresponding to an optimal schedule. Resource allocation has to be solved for all the possible sequences in the POS to find the best solution which is then an optimal schedule. Since the cardinality of POS grows exponentially with the number of jobs, researchers tended to resolve the jobs sequencing subproblem by approximation methods. Jozefowska et al. (1998) developed three metaheuristics: the tabu search, the simulated annealing and the genetic algorithm to solve the sequencing problem. According to the computational experiments results, the performance of tabu search algorithm is more competitive than the remaining two methods. Because of this reason, tabu search algorithm was mainly chosen to be favourably researched and developed (Gupta et al., 2002; Jozefowska et al., 2002b; Waligra, 2009). Since the optimal allocation of a continuous resource of a given jobs sequence was previously calculated by solving non-linear equations, it was nearly incapable to face large problems. Thus, heuristics are additionally introduced trying to find semi-optimal scheduling with a given feasible sequence (Jozefowska et al., 2002a; Waligra, 2010). Recently, Gorczyca et al. (2015) considered a 2-stages hybrid flow shop makespan optimization where jobs have to be preprocessed (can be in parallel) on a single machine with an additional resource. In consequence, the release date of job to the main process is dynamic. The authors proved that this problem is NP-Hard and proposed two local search metaheuristics to solve this one. In the recent years, DCSP has been mainly studied in the computing domain, known as power-aware scheduling problems (Brinkmann et al., 2014; Rozycki and Weglarz, 2012; 2015). Rozycki and Weglarz (2012) have proposed a general methodology for solving this problem in the case that jobs are preemptable and having deadlines. They also provided exact approaches (Rozycki and Weglarz, 2015).

Study motivation and contributions Our study is motivated by a real-life problem: the electric vehicle charging coordination (EVCC). We tend to design a well-performed and fast scheduling algorithm to cope with the real-time scheduling of EVCC. The study in this paper insists on a specific parallel machine problem with a single additional resource where tasks have similar linear resource consumption function $f_i = h_i.u_i$. With the previous literature review, one can see three knowledge gaps in the field, which also highlight the novelty of our work. First, as far as we know, related works have only taken the available amount of additional resource as a constant during the whole scheduling horizon. In this case, the resource allocation of a given job is not necessary to change at every decision interval but only at the completion of other jobs Sadykov (2012). On the contrary, if total available resource varies over times then the resource consumption should also vary at every decision-interval. Second, one have to restrict the increment of this resource allocation between two consecutive intervals. For instance, if a vehicle is currently charging at 3 kW, it cannot charge at 30 kW in the next five-minute due to technical constraints. Third, each job has two restrictions: the time-restriction and the resource consumption restriction. A job's processing is limited between a release date and a deadline. Also, once it starts to process, the resource consumption is bounded between a minimum value and a maximum value. According to our knowledge, with those constraints, our problem becomes one of the most general configurations of the scheduling problem with a single additional resource.

First, we propose an MILP formulation of the problem. We use solver CPLEX to solve the MILP model of our test instances to acquire optimal value. Thus, we can evaluate the performance of our heuristic. This approximate resolution approach for the problem is a constructive heuristic inspired by geometrical strip packing principles. Given a set of semi-flexible item to enter a varying-height strip, first, we try to push all items in to assure feasibility. Then the heuristic tries to polish the first packing solution by greedily shaking and expanding the flexible form of each item.

Problem complexity Regarding the problem notation, we used the three-fields-notation introduced by Graham et al. (1979), with the expansion for the additional resource problem introduced by Blazewicz et al. (1983) and the extension of β fields for resource consumption function introduced by Kellerer (2008) and Kellerer and Strusevich (2008). Therefore, the problem is noted $P_m|res1\cdot\cdot, lin, \Sigma u_{i,k} < U_k, r_i, d_i|\Sigma C_i$. Concerning the complexity, Yalaoui and Chu (2006) pointed out that the parallel machine scheduling problem (PMSP) with release date and total completion time criterion is more general than PMSP with release date and PMSP with total completion time criterion (even with jobs preemption) which are both NP-Hard. Especially, PMSP with release date minimizing total weighted completion times is NP-Hard in the strong sense (Nessah et al., 2008). In addition, Jedrzejowicz and Skakovski (2014) show that the discrete continuous scheduling problem is at least as hard as the resource constrained project scheduling problem since the existence of an additional resource cannot make the problem any simpler. Thus, $P_m|res1\cdot\cdot, lin, \Sigma u_{i,k} < U_k, r_i, d_i| \Sigma C_i$ problem, which is more general than both DCSP with constant resource limit and PMSP with release time minimizing total completion times, is at least NP-Hard.

The paper is organized as follows. In Section 2, we present a mathematical formulation of the problem followed by a mixed integer linear programming model. Section 3 concerns the conception of the heuristics with its principal components. Computational experiments and results are presented in Section 4. We draw conclusions and mention our perspectives for future works in Section 5.

2. Problem formulation

Given a set of n non-preemptive jobs, $J = \{J_1, J_2, \ldots, J_n\}$ to process on m machines $M = \{M_1, M_2, \ldots, M_m\}$ on a scheduling horizon of H decision intervals, each of them lasts Δt units of time. We consider in this paper a time indexed scheduling problem, so decisions should be made only in the beginning of each interval, indexed by interval time $k \in \{0, \ldots, H\}$ (see Fig. 1). Each job has a release date r_i when it arrives into the system, and a deadline d_i, when it has to leave; $r_i \in \{0, \ldots, H\}$ and $d_i \in \{0, \ldots, H\}$. Between two interval times $[k, k+1]$ the processing of job i consumes continuously an amount of resource decided by $u_{i,k}$, s.t. $u_{i,k} \in [u_i^{\min}, u_i^{\max}] \cup \{0\}$ and $\sum_{i=1}^{n} u_{i,k} \leq U_k$, which is called resource consumption capacity constraint. Precisely, $u_{i,k} = 0$ when a job is not in process at time k. In the opposite case, $u_i^{\min} \leq u_{i,k} \leq u_i^{\max}$. The time indexed model is chosen due to the varying of the total available resource.

The state of job i at decision time k is given by $x_{i,k}$, which is the measure of the processed portion of work up to the end of