Discrete Optimization

A constraint generation approach for two-machine shop problems with jobs selection

Federico Della Croce\textsuperscript{a,b,*}, Christos Koulamas\textsuperscript{c}, Vincent T’kindt\textsuperscript{d}

\textsuperscript{a}D.A.I., Politecnico di Torino, Italy
\textsuperscript{b}CNR, IEIIT, Torino, Italy
\textsuperscript{c}College of Business, Florida International University, USA
\textsuperscript{d}Universit\`e Francois-Rabelais, CNRS, LI EA 6300, GC ERL CNRS 6305, Tours, France

\begin{abstract}
We consider job selection problems in two-stage flow shops and job shops. The aim is to select the best job subset with a given cardinality to minimize the makespan. These problems are known to be ordinary NP-hard and the current state of the art algorithms can solve flow shop problems with up to 3000 jobs. We introduce a constraint generation approach to the integer linear programming (ILP) formulation of these problems according to which the constraints associated with nearly all potential critical paths are relaxed and then only the ones violated by the relaxed solution are sequentially reinstated. The proposed approach is capable of solving problems with up to 100,000 jobs.
\end{abstract}

1. Introduction

We consider job selection problems in two-stage shops according to the following specifications. There is a set of $n$ jobs available at time zero; each job $j$ must be processed non-preemptively on two continuously available machines $M_1$, $M_2$ with known integer processing times $a_j$, $b_j$, respectively. Each machine can process at most one job at a time and the operations of each job cannot overlap. The shop is available for some amount of time $d$, the goal is to minimize the number of rejected (selected) jobs to be completed while the shop is available and also to minimize the length of the shop availability period. This is a bi-objective scheduling problem where a natural goal is to search for the set of non-dominated (pareto) solutions.

Since the jobs cannot be preempted, the shop manager should ensure that an appropriate subset of the available jobs is selected each day in order to maximize the number of jobs completed before the shop closes. A practical application of our model is to enable the shop manager to select balanced daily schedules with maximum utility. The schedules will be balanced because the length of the daily availability period is the same for all days in our model. The daily schedules will also be maximum utility schedules because of the objective to maximize the number of jobs. It is implicitly assumed that all jobs have comparable importance and correspondingly equal weights. By doing this, the ranking of the jobs according to their impact on the makespan is facilitated. This enables the shop manager to process the small jobs first so that the number of jobs in-process is minimized and the job delivery rate is maximized.

If the jobs have unequal weights, then a more appropriate objective is the minimization of a composite total cost function such as $TC = C_{\text{max}} + \sum j \in S w_j$ where $C_{\text{max}}$ denotes the makespan of the processed jobs $j \in S$ and $\sum j \in S w_j$ is the total weight of the non-processed jobs; the non-processed jobs are usually outsourced and the weight of a job reflects its outsourcing cost. Choi and Chung (2011) showed that the $F2||C_{\text{max}} + \sum j \in S w_j$ problem is ordinary NP-hard even with specially structured (ordered) job processing times. Lee and Choi (2011) analyzed the $F2||C_{\text{max}} + \sum j \in S w_j$ problem when the outsourcing of an individual operation of a job is allowed so that only the first (second) operation of a job may be processed in-house. The authors showed that the $F2||C_{\text{max}} + \sum j \in S w_j$ problem is ordinary NP-hard with individual operation outsourcing even when each job has the same processing time on both machines and the unit outsourcing cost is machine-dependent but job independent.

If the importance of the jobs is not comparable, then a job-specific rejection cost should be assigned to each job and the job selection problem becomes the corresponding scheduling problem with job rejection. Henceforth, job selection problems can be seen as special cases of scheduling problems with job rejection. We show in Section 3.4 that our solution approach can also handle the case with distinct job weights. The literature on shop scheduling
problems with job rejection is reviewed by Shabtay, Gasper, and Kaspi (2013). The case of the two-machine flow shop problem has been reviewed by Shabtay and Gasper (2012) while T’kindt and Della Croce (2012) provided links with common due date problems. It is mentioned there that most two-stage shop problems with job rejection are NP-hard even with the equal job rejection cost assumption. Zhang, Lu, and Li (2016) presented additional new results for the two-machine flow shop scheduling problem with rejection; they presented more specific versions of the problem that are NP-hard and additional polynomially solvable cases.

In the scheduling literature, several variants of the problem have been considered with different shop configurations. Here, we will consider primarily flow shops and job shops. In the flow shop the order of processing is $M_1 \rightarrow M_2$ for all jobs, while in the job shop there are two job subsets and each job is restricted to having exactly two operations; the jobs in one job subset follow the $M_1 \rightarrow M_2$ processing route while the jobs in the other job subset follow the $M_2 \rightarrow M_1$ processing route. When the availability period $d$ is preset, using the general three-field notation (Lawler, Lenstra, Kan, & Shmoys, 1993), the flow shop problem is denoted as $F[2|d = d| \sum U_j]$ and the job shop problem is denoted as $J[2|d = d| \sum U_j]$. By using the extended three-field notation of T’kindt and Billaut (2006) for multi-objective scheduling, the related problems are also denoted as $F[2|d = d, unknown ~ d|e(\sum U_j)|d]$ and $J[2|d = d, unknown ~ d|e(\sum U_j)|d]$. From now on, we will denote by $S_2$ the two-machine shop problem where the shop configuration may be both $F_2$ or $J_2$ (flow shop or job shop). When the number of jobs to be rejected is given in advance, the corresponding shop problem is denoted as $S_2|d = d, unknown ~ d|e(\sum U_j)|d]$. Finally, if both the number of rejected jobs and the length of the availability period must be minimized and the goal is to search for the non-dominated solutions, then the corresponding shop problem is denoted as $S_2|d = d, unknown ~ d|d| \sum U_j]$. Jozefowska, Jurisch, and Kubiak (1994) showed that all $S_2|d = d| \sum U_j$ problems are ordinary $NP$-hard using reductions from the partition problem and proposed an $O(n^d^2)$ pseudo-polynomial dynamic programming (DP) algorithm for the $F_2|d = d| \sum U_j$ problem (and for its more general weighted version). They also proposed an $O(n^d)$ DP algorithm for the $J_2|d = d| \sum U_j$ problem. Della Croce, Gupta, and Tadei (2000) proposed a branch and bound algorithm for the $F_2|d = d| \sum U_j$ problem. They also observed that the problem is solvable in $O(n \log n)$ time when the jobs and the machines are both ordered because in that case the problem resembles its single-machine counterpart. The ordering of the machines implies that all jobs have their smallest (largest) processing time on the same machine; the ordering of the jobs implies that if $a_i < a_j$ for any two jobs $i, j$, then $b_i < b_j$ as well. Panwalkar and Koulamas (2012) considered the less-restrictive case of ordered machines (without ordered jobs) and proposed an $O(n^2)$ algorithm for the $F_2|d = d| \sum U_j$ problem.

It is of interest to exploit the relationship between the bi-criteria $S_2|d = d|d| \sum U_j$ problem and the corresponding single-objective $S_2|d = d| \sum U_j$ problem in which the common due date $d$ is given. A solution for the $S_2|d = d|d| \sum U_j$ problem can be used to solve the $S_2|d = d| \sum U_j$ problem for any value of $d$. Alternatively, if the $S_2|d = d| \sum U_j$ problem is $NP$-hard, then the $S_2|d = d|d| \sum U_j = NP$-hard as shown in T’kindt, Croce, and Bouquard (2007) for the flow shop case, but easily extendable to the job shop. T’kindt et al. (2007) showed that the $F_2|d = d|d| \sum U_j$ problem is solvable in $O(n^d \log n)$ time by implementing the DP algorithm for the $F_2|d = d| \sum U_j$ problem where $D$ denotes the makespan of an optimal solution to the corresponding $F_2|C_{\max}$ problem. They also proposed an integer linear programming (ILP) formulation and a branch & bound (B&B) algorithm for the $F_2|d = d, unknown ~ d|e(\sum U_j)|d$ problem capable of solving problems with up to 3000 jobs. Then, by applying an $\epsilon$-constraint approach on the number of tardy jobs that repeatedly solved $O(n)$ instances of the $F_2|d = d, unknown ~ d|e(\sum U_j)|d$ problem, they were able to determine exactly the Pareto frontier for the $F_2|d = d|d| \sum U_j$ problem with up to 500 jobs.

The objective of this paper is to show that the ILP formulations involving knapsack-like constraints (such as the one proposed in T’kindt et al. (2007)) can be efficiently tackled by means of a constraint generation approach. As mentioned in Ben-Ameur and Neto (2006), in order to solve ILP models with a large number of constraints, constraint generation techniques are often used when a relaxation of the formulation containing only a subset of the constraints is first solved. Then a separation procedure is applied which adds to the relaxation any inequality of the formulation that is violated by the current solution. The process is iterated until no violated inequality can be found. In this work, all ILP formulations share the feature of having a linear number of constraints where in most cases only a minimal part of these constraints is necessary to reach the optimal solution of the original problem. The computational results reveal a dramatic performance improvement compared to the literature. The proposed approach is capable of solving problems with up to 100,000 jobs for all shop configurations considered. We remark the simplicity of the approach compared to the one proposed in T’kindt et al. (2007). We also point out that constraint generation approaches have already been applied to machine scheduling problems (see, for instance Detienne, Dauzere-Peres, & Yumga, 2011) but never managed to be so successful compared to other available approaches. Correspondingly, by means of the $\epsilon$-constraint approach the limit size of the $S_2|d = d|d| \sum U_j$ problem instances solvable to optimality is also strongly increased. The results indicate as a byproduct a non-intrusive computational statement in the sense that the $F_2|d = d, unknown ~ d|e(\sum U_j)|d$ problem is solved in practice faster than the $F_2|d = d, unknown ~ d|e(\sum U_j)|d$ problem even though the reverse is true if the two problems are solved using the corresponding DP algorithms.

The rest of the paper is organized as follows. The ILP formulations of the $S_2|d = d, unknown ~ d|e(\sum U_j)|d$ problems are presented in Section 2 together with the constraint generation approach. In Section 2, we also establish that the worst-case error of a relaxed version of the models keeping a minimal number of constraints is small providing an intuition on why the proposed constraint generation approach works well. Computational testing showing the effectiveness of the proposed algorithms is provided in Section 3. In Section 3 it is also shown computationally that the addition of job weights does not make the problem harder. Final remarks are reported in Section 4. A preliminary version of part of our findings was presented in Della Croce, Koulamas, and T’kindt (2014).

2. ILP formulations and constraint generation approach

2.1. The $F_2|d = d, unknown ~ d|e(\sum U_j)|d$ problem

Let $a_i$ ($b_i$) denote the processing time of job $i$ on machine $M_1$ ($M_2$). We recall here the integer programming formulation of the $F_2|d = d, unknown ~ d|e(\sum U_j)|d$ problem (hereafter denoted as $F_2$ for conciseness) proposed in T’kindt et al. (2007) where it is assumed that the jobs are indexed and ordered according to Johnson’s rule (Johnson, 1954) that works as follows.

- Schedule first the jobs with $a_i \leq b_i$ in nondecreasing order of $a_i$ followed by the jobs with $a_i > b_i$ in nonincreasing order of $b_i$.

Notice that, if the set $\Omega$ of the $(n-\epsilon)$ selected jobs is fixed, then the optimal value of the shop availability period $d$ is given by $d = \Omega_{\max}(\Omega)$ where $j$ refers to Johnson’s algorithm. Let $d$ denote the unknown value of the shop availability period and $\epsilon$ denote the
دریافت فوری
متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات