Computers & Operations Research 66 (2016) 130–140

Contents lists available at ScienceDirect

Computers & Operations Research

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

# Shifting bottleneck scheduling for total weighted tardiness minimization—A computational evaluation of subproblem and re-optimization heuristics

R. Braune [a,*], G. Zäpfel [b]

[a] Department of Business Administration, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
[b] Institute for Production and Logistics Management, Johannes Kepler University, Altenbergerstrasse 69, 4040 Linz, Austria

ARTICLE INFO

Available online 6 August 2015

Keywords:
Job shop scheduling
Total weighted tardiness
Shifting bottleneck procedure
Decomposition
Dominance rule
Iterated local search

ABSTRACT

Machine-based decomposition of total weighted tardiness job shops is known to be considerably more complicated than in the makespan case, mainly due to the structure of the underlying graph model and thus the arising one-machine subproblems. In fact, the effectiveness of a shifting bottleneck approach crucially depends on the employed subproblem solver. Although a sophisticated exact algorithm exists, problem instances involving more than 30 jobs are still challenging. In this paper, new heuristic approaches to subproblems of this kind are devised which rely on advanced problem-specific concepts like local optimality and dominance principles. The proposed subproblem solvers are combined with an iterated local search method for re-optimizing already scheduled machines. Computational experiments show that the final enhanced shifting bottleneck algorithms are not only applicable to job shops involving up to 100 jobs and 20 machines, but also able to improve existing results for benchmark instances.



## 1. Introduction

The job shop scheduling problem [1] in its standard form is specified by a set of jobs $J = \{1, \ldots, n\}$ to be processed on a shop floor. The shop floor contains a set of machines $M = \{1, \ldots, m\}$, each of which has to be passed exactly once by each job. Accordingly, jobs are split into separate operations with a fixed machine assignment: $o_{ik}$ denotes the operation of job $i$ to be processed on machine $k$ and $p_{ik}$ its positive integer processing time. The technological order of machines, also called the *routing*, is predefined and may differ from job to job. Each machine can only process one job at a time and no preemption is allowed.

A *feasible* schedule is represented by a set of integral operation starting times such that precedence and capacity constraints are satisfied. Given a feasible schedule, let $C_i$ denote the completion time of job $i$, that is, the completion time of the last operation of the job. Then the tardiness $T_i$ of job $i$ with respect to its due date $d_i$ is computed as $T_i = \max(0, C_i - d_i)$. Given job weights $w_i$, the total weighted tardiness (TWT) is then simply equal to $\sum_{i=1}^{n} w_i T_i$. Using the common three-field notation of [2], the total weighted tardiness job shop scheduling problem is written as $Jm||\sum w_i T_i$.

Minimizing total weighted tardiness on a single machine is known to be NP-hard in the strong sense [3], hence $Jm||\sum w_i T_i$ as a generalization of the latter is also classified as strongly NP-hard.

The range of available scientific literature on total weighted tardiness job shops is quite limited compared to the classic makespan ($C_{\max}$) problem. Singer and Pinedo [4] proposed a dedicated branch-and-bound algorithm, which remained the only exact approach in this area until now. Later, the same authors presented a shifting bottleneck procedure (SBP), incorporating problem-specific concepts [5]. The same method also serves as a sub-component of a time-window based decomposition approach for large TWT job shops [6]. The shifting bottleneck paradigm has also been applied for more specialized job shop models, as arising from semiconductor manufacturing scenarios: Mason et al. [7] develop a modified shifting bottleneck procedure for complex job shops involving additional characteristics like sequence-dependent setup times and parallel machines. Mönch and Driessel [8] and Mönch et al. [9] investigate a similar kind of problem in this context. Mönch and Zimmermann [10] study the performance of a shifting bottleneck heuristic under stochastic settings in a semiconductor manufacturing environment.

All of the above approaches specifically address the total weighted tardiness performance measure. However, they considerably differ as far as subproblem solving is concerned. A simple priority rule based approach is adopted by Mason et al. [7], while Mönch et al. [9] apply a genetic algorithm based subproblem solver. Scholz-Reiter et al. [11] as well as Bilyk and Mönch [12] propose a variable neighborhood search algorithm for that purpose. Pinedo and Singer [5] rely on an

* Corresponding author.
E-mail address: roland.braune@univie.ac.at (R. Braune).

http://dx.doi.org/10.1016/j.cor.2015.07.012
0305-0548/

enumeration algorithm of branch-and-bound type for solving single machine subproblems arising from $Jm||\sum w_i T_i$. Bülbül [13] recently proposed a relaxation-based technique for this purpose and integrated it into a shifting bottleneck procedure hybridized with a tabu search heuristic. Braune et al. [14] present a new exact subproblem solver built upon sophisticated problem-specific concepts, particularly dominance rules and lower bounds.

Local search based methods are the second important group of approaches for total weighted tardiness job shops. Singer [15] introduced a cluster-based neighborhood search method. An iterated local search (ILS) algorithm based on the reversal of critical arcs in the graph representation of solutions has been developed by Kreipl [16]. He considers composite moves consisting of at most three related swaps of adjacent operations according to the scheme of Suh [17]. De Bontridder [18] presents a tabu search algorithm for a generalized job shop based on a maximum cost flow model. Relevant work of the recent past includes the genetic local search algorithm of Essafi et al. [19] and the more general local search framework proposed by Mati et al. [20]. A further hybrid genetic algorithm is described in [21]. Kuhpfahl and Bierwirth [22] perform a comprehensive computational comparison of neighborhood structures in the total weighted tardiness context. A simulated annealing approach has been presented by Zhang and Wu [23], while Braune et al. [24] apply an iterated local search algorithm based on advanced approximate move evaluation concepts.

In this paper, several new shifting bottleneck procedures for $Jm||\sum w_i T_i$ are presented. All of them rely on a newly proposed heuristic subproblem solving approach which incorporates a considerable amount of problem-specific knowledge. A dominance rule known from the recently published exact subproblem solver [14] is embedded into a systematic improvement scheme for (partial) operation sequences. The sequence improver is then "plugged" into a list scheduling algorithm based on priority dispatching rules. Hence, the resulting procedure can be considered a *dominance-based heuristic* [25,26].

While the proposed heuristic is able to deliver near-optimal solutions, it is still advisable to use the exact subproblem solver in some situations. The impact on solution quality and runtime behavior of the superordinate bottleneck scheduler is quantified in an experimental way, based on benchmark instances of job shop type.

The different variants of the proposed bottleneck approach emerge from alternative re-optimization schemes. Besides the conventional, single-machine oriented re-optimization, an iterated local search (ILS) algorithm is employed for improving partial solutions at the job shop level. It is actually used in two configurations: (1) As a standalone re-optimizer and (2) in combination with conventional re-optimization.

The paper is organized as follows: Section 2 describes the disjunctive graph model which serves as the basis for bottleneck scheduling and all related activities. Section 3 gives a brief overview of the shifting bottleneck procedure itself, followed by a detailed coverage of the two main topics of the paper, subproblem solving (cf. Section 4) and re-optimization (cf. Section 5). Comprehensive computational experiments and their discussion are the contents of Section 6. The final Section 7 provides concluding remarks and outlooks on future research.

## 2. Disjunctive graph model

In the job shop case, a disjunctive graph [27] can be defined as a triple $G=(V,W,Z)$. $V$ denotes the set of all nodes (vertices), which encompasses not only the ones corresponding to operations but also a source node $X$ and, depending on the objective function, one or more sink nodes. The total weighted tardiness as a so called "min-sum" objective in fact requires the incorporation of $n$ sink nodes $Y_1, \dots, Y_n$, one for each job (cf. Fig. 1). The precedence constraints between operations of the same job are reflected by directed edges included in set $W$. $Z$ denotes the set of undirected or disjunctive arcs between operations on the same machine.

The graph model considered in this paper is edge-weighted, hence the weight or length of each edge corresponds to the processing time of the operation node it originates from. The length of an edge starting from the source node $X$ corresponds to the release time of the associated job.

A schedule can be obtained by orienting the disjunctive edges between operations on the same machine. This is achieved by a *selection* of arcs from the set $Z$. The solution is feasible, if the resulting digraph does not contain any cycle. Let $L(u,v)$ denote the length of a longest path between two arbitrary nodes $u$ and $v$ in the graph. If no such path exists, then $L(u,v)$ is undefined. The completion time $C_i$ of a particular job is exactly the length of a longest path from the source $X$ to sink $Y_i$, i.e., $L(X, Y_i)$.

## 3. Shifting bottleneck framework

The main steps of a shifting bottleneck procedure (SBP) as initially proposed by Adams et al. [28] for the makespan objective can be considered in an abstract way and thus independent of a particular objective. Fig. 2 provides a flow chart representation of the basic framework of a bottleneck scheduling algorithm. Note that $M$ refers to the set of all machines while $M_0$ denotes the set of already scheduled machines which is continuously updated during the run of the algorithm.

Pinedo [5] and Bülbül [13] have already proposed custom implementations of this generic SBP, which are tailored to the total weighted tardiness objective.

This paper presents conceptual innovations and enhancements with regard to the subproblem solving subroutine, for which new heuristic approaches are devised (cf. Section 4). Apart from that, two alternative re-optimization methods are proposed (cf. Section 5). In the standard case, re-optimization of the current partial solution is accomplished by re-sequencing already scheduled machines using the single machine subproblem solver. However, a partial solution at the job shop level may also be improved by any local search algorithm which is based on the graph representation (cf. Section 2). Other than Bülbül [13], who integrates a tabu search algorithm, an even more simple local search algorithm is employed in this paper and finally also combined with the conventional re-optimization approach (cf. Section 5.2).

The order in which machines are processed depends on the applied bottleneck criterion. A common approach is to first schedule the machine causing the highest increase of the objective function value. However, slight variations of the sequence of processed machines may significantly improve the solution

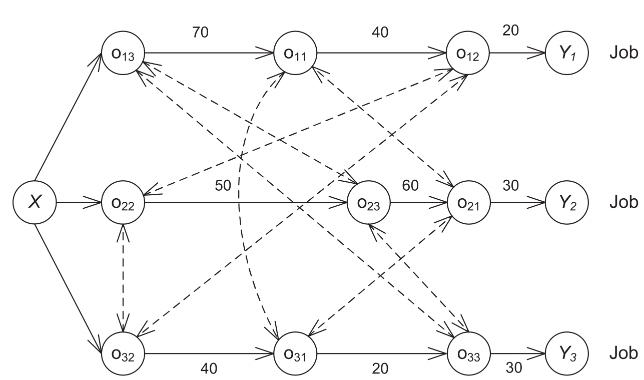


**Fig. 1.** Disjunctive graph model for a 3 × 3 job shop given a min-sum objective.