An effective lower bound on $L_{\text{max}}$ in a worker-constrained job shop

Benjamin J. Lobo\textsuperscript{a}, Thom J. Hodgson\textsuperscript{a}, Russell E. King\textsuperscript{a}, Kristin A. Thoney\textsuperscript{b}, James R. Wilson\textsuperscript{a}

\textsuperscript{a}Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Campus Box 7906, Raleigh, NC 27695-7906, USA
\textsuperscript{b}Department of Textile and Apparel Technology and Management, North Carolina State University, Campus Box 8301, Raleigh, NC 27695-8301, USA

Abstract

A common industrial operation is a dual resource constrained job shop where: (a) the objective is to minimize $L_{\text{max}}$, the maximum job lateness; (b) machines are organized into groups; and (c) each worker is assigned to a specific machine group. Because this problem is NP-hard, finding optimal solutions by enumeration is impractical. This article details a procedure to compute a lower bound on $L_{\text{max}}$ that will be used in follow-up work to effectively evaluate the absolute performance of heuristic solutions. Given an allocation of workers to machine groups, a lower bound on $L_{\text{max}}$ is first computed for each machine group using a network-flow formulation. The lower bound on $L_{\text{max}}$ for the job shop is the largest of the lower bounds for the machine groups. A search algorithm then finds a worker allocation yielding the smallest such lower bound on $L_{\text{max}}$ for the job shop; and the latter quantity is our proposed lower bound on $L_{\text{max}}$. Given a worker allocation, we use the Virtual Factory (a heuristic scheduler developed by Hodgson et al. in 1998) to generate a schedule. Experiments with a wide variety of job shops indicated that the proposed lower bound on $L_{\text{max}}$ could often be achieved by a Virtual Factory schedule based on the worker allocation yielding this lower bound. However, there were problem instances for which other worker allocations enabled the Virtual Factory to generate better schedules. Follow-up work provides optimality criteria, and heuristics to find improved allocations if these criteria are not satisfied.

Keywords: job shop scheduling, dual resource constrained systems, maximum lateness, worker allocation

1. Problem Definition

Most job shop scheduling problems have been shown to be NP-Hard [1], and as a result there has been considerable effort spent researching better methods of finding optimal or near-optimal solutions (see [2, 3, 4]). Traditionally, job shop scheduling has focused on the case in which the job shop is constrained only by the number of machines in it. Determination of a schedule that minimizes $L_{\text{max}}$, the maximum job lateness, in a job shop with $N$ jobs and $M$ machines is NP-Hard [5]. The basic assumptions are that:

- Each job has a specific routing through the machines.
- There are no machine breakdowns.
- No preemption is allowed.
- Transportation time between different machines and set-up time on machines is negligible (one can view the processing time of a job as already having the set-up time included).
- A machine can only process one job at a time, and a job can only be processed on one machine at a time.
Implicit in these assumptions is that there is always a worker available to run the machines, i.e., every machine can be operated at all times. However, in a typical job shop setting, the number of workers available to operate the machines is often far less than the number of machines present in the job shop [6]. Since machines cannot process jobs if they are not being operated, workers are often moved among the various departments to perform production operations in order to better meet due-dates. We have observed this phenomenon typically where work forces are mature and workers have the necessary skills to perform multiple process operations.

Systems where both machines and workers are treated as constraints are termed dual resource constrained (DRC) systems [7, 8]. The importance of DRC systems is evidenced by the substantial literature addressing such problems (see the reviews by Treleven [7], Gargeya and Deane [6], and Hottenstein and Bowman [9]). Starting with Nelson [10], research has focused on how the interaction between the two constrained resources affects job shop performance, but has varied widely in terms of the assumptions made about the second constrained resource (e.g., worker skill level, efficiency, training). The addition of workers to the $J_M || L_{\max}$ job shop scheduling problem (a job shop with $N$ jobs and $M$ machines in which the objective is the minimization of $L_{\max}$) makes it even more complex: a solution to this problem must now also specify the group of machines to which each worker is assigned.

Our approach to allocation of fully cross-trained workers was motivated by our examination of, and involvement with, the operations in a major high-end furniture manufacturer, a large-scale apparel producer, and the US Navy’s Aviation Depots. In each case, cross-trained workers were allocated on an ad hoc basis; and in each of these disparate organizations, there was a clearly recognized need for a more systematic and effective approach to the allocation of workers.

According to Adams et al. [11] “job shop scheduling is among the hardest combinatorial optimization problems.” They go on to state that “since job shop scheduling is a very important everyday practical problem, it is therefore natural to look for approximation methods that produce an acceptable solution in useful time.” This article addresses the development of a lower bound to provide an absolute (rather than comparative) basis for evaluating heuristic solutions. A DRC job shop problem is investigated in which machines are grouped into machine groups and each worker is assigned to a specific machine group to operate the machines within it. Together with the traditional job shop assumptions, the following assumptions transform the job shop into a DRC system:

- There are fewer workers than machines in the job shop.
- A machine group must have at least one worker assigned to it, otherwise the machine group cannot process any jobs.
- A machine requires one worker to operate it, and a worker cannot operate more than one machine at once; moreover, the worker must be present at a machine during each time interval in which a job is being processed on that machine.
- The allocation of workers to machine groups, once determined, is static.
- Every worker can be assigned to any machine group in the job shop.

We have observed that, in a large number of manufacturing facilities and maintenance depots, the primary concern is the completion of all jobs by their due dates. This observation has also been made by Ovacik and Uzsoy [12], who find that most companies relate the on-time completion of jobs to the satisfaction of the customer. Therefore the maximum job lateness, $L_{\max}$, is used as the performance measure in this research. The objective is to minimize $L_{\max}$ by making an optimal assignment of workers to machine groups, and by formulating an optimal schedule for processing the set of jobs on their required machines. This objective tends to ensure that all jobs are completed as early as possible relative to their due-dates. In addition, if there
is a schedule that satisfies all customers by having no late jobs (i.e., \( L_{\text{max}} \leq 0 \)), minimizing \( L_{\text{max}} \) addresses the on-time completion directly.

This article is organized as follows. The literature review is presented in Section 2, and in Section 3 an approach to solving the \( J_M || L_{\text{max}} \) single resource constrained job shop scheduling problem is reviewed. Section 4 first presents a procedure to find a lower bound on \( L_{\text{max}} \) given an allocation of workers to machine groups, and then presents an optimal algorithm to find an allocation yielding the smallest such lower bound. The experimental results concerning this allocation are presented in Section 5.

Because this problem is NP–Hard, the bound developed in this article enables absolute (and effective) evaluation of heuristic solutions, and this evaluation is presented in Lobo et al. [13].

2. Literature Review

Dual resource constrained (DRC) systems treat manpower as a constraint [7, 8]. A DRC shop is “one in which shop capacity may be constrained by machine and labor capacity or both. This situation exists in shops that have equipment that is not fully staffed and machine operators who are capable of operating more than one piece of equipment” [8]. In the DRC job shop, the worker “may be transferred from one work centre to another (subject to skills restrictions) as the demand dictates” [8]. Gargeya and Deane [6] note that part of the complexity of scheduling in DRC systems stems from the need for a rule for assigning manpower to the machines.

Treleven [7] reviews the literature on DRC systems, summarizing the various models used, the parameters of the systems investigated, the job dispatching and worker allocation rules employed, and the different criteria by which system performance was evaluated. In their review of studies that involved DRC systems, Hottenstein and Bowman [9] find that there are two main questions regarding the allocation of workers — when should workers transfer from one machine group to another, and to which machine group (where) should they move? They find that the research addresses these questions through investigation of worker flexibility, centralization of control, worker allocation rules, queue discipline, and the cost of transferring workers. More recent research focuses on the effects of the following factors on various performance criteria for DRC job shops: (a) cross-training workers to operate machines in different departments (machine groups); and (b) incorporating more-realistic assumptions about worker behavior—e.g., learning, fatigue, and forgetfulness [14, 15, 16, 17].

Felan et al. [18] look at the effect that labor flexibility and staffing levels have on job shop performance measures. They consider a homogeneous workforce whose flexibility comes from the ability to work in a varying number of departments in the job shop, and where a given staffing level corresponds to each department having the same number of workers assigned to it. They measure the combined effects of labor flexibility and staffing level on work in progress (WIP), due-date performance, and cost criteria; these criteria “represent the primary drivers for measuring manufacturing performance in many organizations” [18]. They find that for a given workforce flexibility level, an increase in the staffing level yields an increase in system cost, a decrease in the WIP and an improvement in the due-date performance. For a given staffing level, an increase in the worker flexibility level also yields an increase in system cost, a decrease in the WIP and an improvement in the due-date performance. They conclude that based on the diminishing returns apparent in the WIP and due-date performance, their optimal job shop has a staffing level of 60% with a workforce that has a medium flexibility level. Although Felan et al. [18] allow workers to transfer between departments, they are concerned with finding the optimal staffing level and workforce flexibility level combination, rather than optimizing the allocation of workers to departments given a staffing level.

The shifting bottleneck heuristic [11] is an iterative procedure for finding a job shop schedule that minimizes the makespan (i.e., the completion time of the last job to leave the system). On each iteration of the heuristic, every machine not in the set \( M_0 \) is considered as a separate \( 1 | r_j | L_{\text{max}} \) scheduling problem that involves only the sequencing constraints for the machines currently in \( M_0 \) and that yields its own minimum
value of $L_{\text{max}}$, where the heuristic has the initial condition $M_0 = \emptyset$. The bottleneck machine corresponds to the largest such value of $L_{\text{max}}$ for the machines not in $M_0$. The bottleneck machine is added to $M_0$; and then each machine already in $M_0$ is again considered as a separate $1|r_j|L_{\text{max}}$ scheduling problem that involves only the sequencing constraints for the other machines currently in $M_0$, including the newly added machine. The heuristic terminates if $M_0$ contains all machines in the job shop; otherwise a new iteration is performed to schedule the next bottleneck machine, add that machine to $M_0$, and then reschedule all the other machines in $M_0$. An in-depth explanation of the shifting bottleneck heuristic can be found in Chapter 7 of Pinedo [19].

The methodology presented in this article differs from the shifting bottleneck procedure in several significant respects. First, we consider a DRC system that is constrained by the availability not only of machines, but also of the workers needed to operate the machines. Second, we allow preemption when determining a bottleneck machine group, a relaxation that enables us to calculate an effective lower bound on $L_{\text{max}}$ for a given allocation of workers to machine groups. Finally, each iteration of the heuristic scheduler uses estimated queuing times of all jobs in the job shop to reoptimize the schedule for all machines simultaneously. By contrast, each iteration of the shifting bottleneck heuristic uses only the job-sequencing constraints for the machines currently in the set $M_0$.

While there has been extensive research into DRC systems, to our knowledge there has not been any published work on the problem addressed in this article—namely, the allocation of homogeneous, fully cross-trained workers in a DRC job shop so as to minimize $L_{\text{max}}$. This article makes the following contributions: (a) a method to compute an effective lower bound on $L_{\text{max}}$ for a given allocation of workers to machine groups; (b) a search algorithm that is guaranteed to find an allocation yielding the smallest such lower bound; and (c) a lower bound on $L_{\text{max}}$ for the $J_M\|W\|L_{\text{max}}$ DRC job shop scheduling problem that enables absolute evaluation of heuristic solutions to this NP-hard problem.

3. A heuristic solution approach to the $J_M\|L_{\text{max}}$ job shop scheduling problem

This section reviews the solution approach developed by Hodgson et al. [20] when solving the $J_M\|L_{\text{max}}$ single resource constrained job shop scheduling problem. The Virtual Factory is used in the current research as a heuristic scheduler to generate schedules given an allocation of workers to machine groups.

Hodgson et al. [20] developed the Virtual Factory as part of their solution approach to the $J_M\|L_{\text{max}}$ single resource constrained job shop scheduling problem. The Virtual Factory is a transient, deterministic simulation of a job shop, where the factory is run from its current state until all jobs are completed. It is based on an approach proposed by Vepsalainen and Morton [21], and at each iteration determines and then simulates an actual schedule for the job shop. In the first iteration, the jobs are ordered on each machine based on slack. However, early experimental tests by Carroll [22] show that slack does not perform well as a sequencing rule, due to the fact that slack does not account for queuing time — the time job $j$ spends waiting in the queue at machine $h$ before processing begins. Therefore, in subsequent iterations, jobs are scheduled using a revised slack measure that takes the queuing time of the jobs (determined from the schedule generated in the previous iteration) into account. If the due-date of job $j$ is denoted as $d_j$, the processing time of job $j$ on machine $h$ is denoted by $p_{jh}$, and the queuing time of job $j$ at machine $h$ is denoted by $q_{jh}$, then the revised slack of job $j$ on machine $h$ is defined as

$$\text{slack}_{jh} = d_j - \sum_{k \in m^+_j} p_{jk} - \sum_{k \in m^{++}_j} q_{jk}, \quad (1)$$

where $m^+_j$ is the set of all operations of job $j$ subsequent to machine $h$, and $m^{++}_j$ is the set of all operations of job $j$ subsequent to machine $h$ except the immediately subsequent operation. The Virtual Factory is run for a specified number of iterations or until the lower bound on $L_{\text{max}}$, the smallest possible value of $L_{\text{max}}$
that can be attained, is achieved. The Virtual Factory has been shown to provide near optimal solutions to industrial-size problems in near real time [20].

Carlier and Pinson [23] detail the method of obtaining the lower bound that is used in the Virtual Factory. The earliest possible start time for each job \( j \) on machine \( h \) is defined as

\[
ES_{jh} = \sum_{k \in m_j}^m p_{jk},
\]

(2)

where \( m_j \) is the set of all operations of job \( j \) previous to machine \( h \). Similarly, the latest possible finish time for each job \( j \) on machine \( h \) is defined as

\[
LF_{jh} = d_j - \sum_{k \in m_j^+} p_{jk}.
\]

(3)

Since \( ES_{jh} \) is the earliest possible time that job \( j \) can get to machine \( h \), the value of \( ES_{jh} \) can be viewed as the effective release time (\( r_{jh} \)) for job \( j \) on machine \( h \); and since \( LF_{jh} \) is the latest possible time job \( j \) can finish on machine \( h \) and still complete processing on time, \( LF_{jh} \) can be viewed as the effective due-date for job \( j \) on machine \( h \).

With the release times and due-dates for every job on every machine computed, it is possible to solve an instance of the \( 1|r_j|\text{max} \) job shop scheduling problem on every machine \( h \) in the job shop. The \( 1|r_j|\text{max} \) job shop scheduling problem is NP-hard and can be solved optimally using branch and bound [24]; however, if preemption of a job in process is allowed whenever another job with a more imminent due-date becomes available, then a lower bound can be computed very quickly using a procedure that schedules the job with the smallest due-date among all jobs that have been released at each point in time [24, 25]. As a result of this relaxation, the lower bound is not necessarily tight. If \( LB_h \) denotes the lower bound on \( L_{\text{max}} \) for the \( 1|r_j|\text{max} \) job shop scheduling problem on machine \( h \), then the lower bound on \( L_{\text{max}} \) for the \( JM|\text{max} \) job shop scheduling problem is taken to be

\[
LB = \max_{1 \leq h \leq M} \{LB_h\}.
\]

Comparing the Virtual Factory solution with \( LB \) gives a valid upper bound on the amount by which the Virtual Factory–generated value of \( L_{\text{max}} \) could exceed the minimum achievable value of \( L_{\text{max}} \). Empirically, \( LB \) appears to be an effective lower bound on \( L_{\text{max}} \) for the overall job shop since there are \( M \) opportunities for it to be tight [20].

Virtual Factory extensions include generating schedules that identify and prioritize critical jobs [26], identifying the best alternative process plans through the incorporation of tabu search [27], modeling multi-factory scenarios [28, 29], optimizing the release time of jobs to the factory floor [30], and incorporating simulated annealing to improve solution quality [31]. In this research the Virtual Factory is used to generate schedules for this NP–Hard problem.

4. A lower bound on \( L_{\text{max}} \) for the DRC job shop scheduling problem

Given a particular allocation \( \vartheta \) of workers to machine groups in the job shop, a lower bound on \( L_{\text{max}} \) for each machine group can be determined. A lower bound on \( L_{\text{max}} \) for the entire DRC job shop given allocation \( \vartheta \), denoted \( LB_{\vartheta} \), is then simply the maximum of the individual machine group lower bounds. A search algorithm can then be used to find an allocation \( \vartheta^* \) of workers to machine groups yielding the smallest value of \( LB_{\vartheta} \) over all feasible allocations \( \vartheta \). The new DRC problem is defined in Section 4.1. Section 4.2 describes how to determine a lower bound on \( L_{\text{max}} \) for a single machine group, while Section 4.3 outlines the search algorithm that finds an allocation of workers to machine groups yielding the smallest value of \( LB_{\vartheta} \) for the DRC job shop.
4.1. Definition of the new DRC problem

As in the \( J_M || L_{\text{max}} \) job shop scheduling problem, the DRC job shop has a total of \( N \) jobs and \( M \) machines on which to process the jobs. For \( j = 1, \ldots, N \), job \( j \) has a fixed route through the job shop, a processing time \( p_{jh} \) at each machine \( h \) that the job visits on its route, and a due-date \( d_j \). Machines in the DRC job shop that perform similar tasks are grouped together, where the \( i \)th machine group is denoted as \( mg_i \) for \( i = 1, \ldots, S \), and \( S \) is the number of machine groups. Although machines are grouped by their ability to perform similar tasks, we assume that each job must be processed on the machine(s) specified on its route; and thus the job shop is not a flexible job shop [19, Chapter 2]).

Remark: In many types of manufacturing facilities, each job is assigned to a specific machine in each machine group visited by the job. For example in a semiconductor wafer fabrication facility, machines are grouped together based on the manufacturing process step (e.g., photolithography) that they perform on wafers organized into lots. At every step in the processing of a wafer lot of each product type, the routing decision requires assignment of the lot to a specific machine (e.g., a specific model or generation of lithography equipment); see Asmundsson et al. [32, pp. 99–100] and Kumar and Kumar [33, p. 549].

Of the \( W \) workers in the job shop, \( w_i \) workers will be assigned to machine group \( i \), where \( w_i \geq 1 \) for \( i = 1, \ldots, S \) and \( \sum_{i=1}^{S} w_i = W \). The objective is to determine an allocation of workers to machine groups so that the schedule with the smallest \( L_{\text{max}} \) value can be generated. This problem is defined as the \( J_M | W | L_{\text{max}} \) DRC job shop scheduling problem, where \( W \) indicates the dual constraint imposed by workers.

Suppose we have obtained a particular allocation \( \vartheta = [w_1, \ldots, w_S] \) of workers. Since the routes of all jobs in the job shop are known, we can determine which jobs pass through machine group \( i \) and, furthermore, on what machine(s) in machine group \( i \) each job must be processed. We can then define a \( J_{M_i}[w_i, r_{jh}] | L_{\text{max}} \) DRC job shop scheduling subproblem for each machine group \( i \) if we set \( N_i = |N_i| \) where \( N_i \) is the set of all jobs that are routed through a machine in machine group \( i \), \( M_i = |M_i| \) where \( M_i \) is the set of machines in machine group \( i \), \( 1 \leq w_i \leq |M_i| \), and \( r_{jh} \) are the local release times. For machine group \( i \), the set \( \delta_{ih} \) is the set of job operations that must be processed on machine \( h \in M_i \). Because a job can have more than one operation occur in the same machine group, it is the case that

\[
\sum_{h \in M_i} |\delta_{ih}| \geq |N_i|.
\]

The local release time \( r_{jh} \) for job \( j \) on machine \( h \in M_i \) is the earliest possible start time of that job on machine \( h \), and is computed using Equation (2). The local due-date for a job on machine \( h \in M_i \) is the latest possible finish time of that job on machine \( h \), and is computed using Equation (3).

Given a particular allocation \( \vartheta \) of workers to the \( S \) machine groups, the solution to the preemptive \( J_{M_i}[w_i, r_{jh}] | L_{\text{max}} \) DRC job shop scheduling subproblem (described in Section 4.2) provides a lower bound on \( L_{\text{max}} \) for the \( i \)th machine group. Let \( w_i(\vartheta) \) denote the number of workers assigned to machine group \( i \) under allocation \( \vartheta \), and let \( LB_{mg_i}(w_i(\vartheta)) \) denote the lower bound on \( L_{\text{max}} \) for machine group \( i \) based on the local due-dates for jobs visiting machines in machine group \( i \) and given that \( w_i \) workers are assigned to it under allocation \( \vartheta \). For a particular allocation \( \vartheta \), the lower bound on \( L_{\text{max}} \) for the entire job shop is

\[
LB_{\vartheta} = \max_{1 \leq i \leq S} \{LB_{mg_i}(w_i(\vartheta))\}.
\]

In addition, there exists an allocation \( \vartheta^* \) (not necessarily unique) for which it is the case that

\[
LB_{\vartheta^*} \leq LB_{\vartheta}
\]

for every other feasible allocation \( \vartheta \) of workers to the machine groups. For a particular instance of the \( J_M | W | L_{\text{max}} \) DRC job shop scheduling problem, Section 4.3 describes a method for finding an allocation \( \vartheta^* \) that satisfies Equation (4), i.e., an allocation \( \vartheta^* \) yielding the smallest \( LB_{\vartheta} \) value over all feasible allocations.
4.2. Finding the lower bound on $L_{\text{max}}$ for a machine group

4.2.1. A network flow formulation for a machine group

The problem of finding $LB_{\text{mg}}(w_i(\vartheta))$ — the lower bound on $L_{\text{max}}$ for machine group $i$ based on the local due-dates for jobs visiting machines in machine group $i$ and given that $w_i$ workers are allocated to it under allocation $\vartheta$ — can be formulated as a series of network flow problems whose general structure is given in Figure 1. A trial value of $L_{\text{max}}$, denoted as $y$, must be assumed for each network flow formulation, and the feasibility of the network depends upon the value of $y$ chosen. The procedure presented here is an extension of an approach presented by Labetoulle et al. [34] for finding the optimal solution to the preemptive parallel machine problem with the objective of minimizing $L_{\text{max}}$. The primary difference in our network is the addition of a worker node for each time period that constrains the number of machines that can be operated in a given time period to the number of workers assigned to the machine group.

In Figure 1 the source node for machine group $i$ is labeled $s_i$. The sink node is labeled $k_i$. The job operation nodes of machine $h \in M_i$ are labeled $j_{u,h}$ where $u \in \mathcal{J}_{ih}$. The time period nodes are labeled $t_{v,h}$ and represent the interval from time $v$ to time $v + 1$. The integer-valued time $v$ can take on values ranging from time $\delta_h$ to time $\tau_h$. The variable $\delta_h$ is the earliest point in time that processing can start on machine $h$, while given the value $y$, all processing on machine $h$ must be finished by time $\tau_h + 1$. The value of $\delta_h$ is calculated as

$$\delta_h = \min_{w \in \mathcal{J}_{ih}} \{ES_{uh}\}$$

for $h \in M_i$, where $ES_{uh}$ is earliest start time of job operation $u$ on machine $h$ (computed using Equation (2)). The value of $\tau_h$ is calculated as

$$\tau_h = \max_{w \in \mathcal{J}_{ih}} \{LF_{uh}\} + y - 1$$

for $h \in M_i$, where $LF_{uh}$ is the latest finish time of job operation $u$ on machine $h$ (computed using Equation (3)).

The worker nodes for the $i$th machine group are labeled $\omega_{i,g}$ and represent the interval from time $g$ to time $g + 1$. The integer-valued time $g$ can take on values ranging from time $\psi$ to time $T$. The variable $\psi$ is the earliest point in time that processing can start on any machine $h$ in machine group $i$, while given the value $y$, all processing on any machine $h$ in machine group $i$ must be finished by time $T + 1$. The value of $\psi$ is calculated as

$$\psi = \min_{h \in M_i} \{\delta_h\},$$

and the value of $T$ is calculated as

$$T = \max_{h \in M_i} \{\tau_h\}.$$

For each machine $h \in M_i$, there is an arc from node $s_i$ to node $j_{u,h}$ for $u \in \mathcal{J}_{ih}$. As shown in Figure 2, there is an arc from node $j_{u,h}$ to nodes $\{t_{v,h} : v = \alpha, \alpha + 1, \ldots, \beta\}$, where

$$\alpha = ES_{uh} \quad \text{and} \quad \beta = LF_{uh} + y - 1.$$

Referring back to Figure 1, for each machine $h$, we see that there is an arc from node $t_{v,h}$ to the corresponding worker node $\omega_{i,v}$ for $v = \delta_h, \ldots, \tau_h$. There is also an arc from node $\omega_{i,g}$ to node $k_i$ for each $g = \psi, \ldots, T$.

An arc is specified as a (From, To) node pair, and the capacities on an arc (given in Table 1) are specified as a [Lower Bound, Upper Bound] pair. The capacities ensure that

(i) For each job operation $u$ on machine $h \in M_i$, there must be a total flow of $p_{uh}$ units of processing time (i.e., the entire job’s operation must be processed),
(ii) For each job operation $u \in \mathcal{J}_i$ at most one unit of processing time can be completed during any time period $t$.

In addition, the capacities ensure that during each unit time period, at most $w_i$ machines can be operated (corresponding to the $w_i$ workers allocated to machine group $i$). Because the network flow formulation allows preemption, the smallest value of $y$ that allows a feasible network is actually a lower bound on $L_{\text{max}}$ for machine group $i$ given that there are $w_i$ workers allocated to it, i.e., $\text{LB}_{\text{mg}i}(w_i(\bar{\theta}))$.

Table 1: Arc capacity lower and upper bounds for machine group $i$.

<table>
<thead>
<tr>
<th>Arc</th>
<th>Capacity</th>
<th>Relevant Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(s_i, j_{uh})$</td>
<td>$[P_{uh}, P_{uh}]$</td>
<td>$u \in \mathcal{J}_{ih}, \ h \in \mathcal{M}_i$</td>
</tr>
<tr>
<td>$(j_{uh}, t_{vh})$</td>
<td>$[0, 1]$</td>
<td>$u \in \mathcal{J}_{ih}, \ v = \delta_h, \ldots, \tau_h, \ h \in \mathcal{M}_i$</td>
</tr>
<tr>
<td>$(t_{vh}, \omega_{i,v})$</td>
<td>$[0, 1]$</td>
<td>$v = \delta_h, \ldots, \tau_h, \ h \in \mathcal{M}_i$</td>
</tr>
<tr>
<td>$(\omega_{i,g}, k_i)$</td>
<td>$[0, w_i]$</td>
<td>$g = \psi, \ldots, T$</td>
</tr>
</tbody>
</table>

The smallest value of $y$ that allows a feasible network can be determined in the following way. Assume that a given value of $y$ is known to allow a feasible network. A series of network flow problems can then be solved, where the value of $y$ for each successive network is decreased by one unit until the first infeasible network is found. The out-of-kilter algorithm can detect infeasibility in a network flow problem [35], but to use this algorithm, an initial feasible $y$ value must be obtained.
4.2.2. Finding an initial value for $y$

To find a $y$ value that allows a feasible network, we initially assume that a worker is always available to operate a machine. Since we know exactly what jobs must be processed in the machine group, a $1|\text{\emph{r}_{\text{fh}}}|L_{\text{\emph{max}}}$ job shop scheduling problem with preemption can be solved for each machine $h \in \mathcal{M}_i$ (where $N = |\mathcal{J}_i|$).

Since there are only $w_i$ workers in machine group $i$, where $1 \leq w_i \leq |\mathcal{M}_i|$, the assumption that each machine can be operated when necessary may not be true. If there is a point in time $t$ at which more than $w_i$ machines must be operated simultaneously, then the schedules as determined by solving the $|\mathcal{M}_i|$ different $1|\text{\emph{r}_{\text{fh}}}|L_{\text{\emph{max}}}$ job shop scheduling problems with preemption are infeasible and must be modified, and the decision must be made about which $w_i$ machines to operate and which of the $|\mathcal{M}_i|$ machines to not operate.

At time $t$, nothing can be done to change the lateness values of the jobs scheduled before time $t$, so the only consideration is what to do with the jobs that remain to be scheduled at time $t$. Since the lateness value of any one of the remaining jobs on a machine has the potential to yield the $L_{\text{\emph{max}}}$ value for the entire machine group, the objective at time $t$ is to schedule the remaining jobs such that the $L_{\text{\emph{max}}}$ value for each machine remains the same as it was at time $t$. A schedule built by operating the $w_i$ machines with the largest $L_{\text{\emph{max}}}$ values (from the $|\mathcal{M}_i|$ schedules generated from the jobs remaining at time $t$) attempts to minimize $L_{\text{\emph{max}}}$ for the machine group.

Algorithm 1, which aims to minimize $L_{\text{\emph{max}}}$, is a constructive algorithm that takes worker availability into account when generating a preemptive machine group schedule for machine group $i$. Although the value of $L_{\text{\emph{max}}}$ provided by this constructive algorithm is not necessarily the smallest $y$ value that allows a feasible network, it does allow a feasible network. In practice the initial $y$ value provided by Algorithm 1 is typically very close to the final $y$ value; the difference between the initial and final values of $y$ was less than or equal to 1 in over 80% of the 161,427 times that the network flow approach was required in the experiments for this article. In addition, the difference between the initial and final values of $y$ was at most 5 in over 98% of the times the network flow approach was required for the experiments reported in this article.
Algorithm 1 Constructive algorithm for determining an initial $y$ value for $mg_i$.

$t \leftarrow 0.$
$\Psi \leftarrow M_i$

1. **For each** machine $h \in \Psi$
   Generate the optimal preemptive schedule from time $t$ onwards for machine $h$, assuming a worker is always available when needed.

2. $m \leftarrow 0.$
   **For each** machine $h \in M_i$
   
   - If machine $h$ needs to be operated at time $t$
     
     $m \leftarrow m + 1$
   
   If $m > w_i$
   
   Determine the $w_i$ machines with the $w_i$ largest $L_{\max}$ values.
   Schedule the $w_i$ machines to be operated at time $t$.
   Let $\Psi$ be the set of all machines that needed to be operated at time $t$ but were not.
   $t \leftarrow t + 1$
   **Go to 1.**

   Else
   
   Schedule all machines that need to be operated at time $t$.
   If No jobs remain on any machine
   
   $y \leftarrow \max_{h \in M_i} \{LB_h\}$
   
   **RETURN** $y$

   Else
   
   $t \leftarrow t + 1$
   **Go to 2.**


4.2.3. Characterization of the lower bound on $L_{\text{max}}$ for a machine group

The lower bound on $L_{\text{max}}$ for a machine group is a discrete function of the number of workers assigned to the machine group. The following example illustrates that the addition of one or more workers to a machine group can improve the lower bound on $L_{\text{max}}$ for the machine group. Consider a machine group that has seven different machines with seven different jobs to be processed, one on each machine. Each job has a release time of 0, a due-date of 0, and a processing time of 2. The lower bound on $L_{\text{max}}$ values as a function of the number of workers assigned to the machine group are given in Table 2. We present the computation of the lower bound on $L_{\text{max}}$ given that two workers are assigned to the machine group. Without loss of generality, assume that starting at time 0 the first worker processes job 1, job 2, and then job 3, while starting at time 0 the second worker processes job 4 and then job 5. The second worker processes job 6 from time 4 to time 5, at which point job 6 is preempted by job 7 which the second worker finishes processing at time 7. At time 6 the first worker is done with job 3 and resumes job 6, finishing processing at time 7. The lower bound on $L_{\text{max}}$ when more workers are assigned to the machine group can be computed similarly. From the table it is clear that increasing the number of workers assigned to a machine group can improve the lower bound on $L_{\text{max}}$.

Table 2: Lower bound on $L_{\text{max}}$ values for a machine group as a function of the number of workers assigned

<table>
<thead>
<tr>
<th>Number of Workers Assigned</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB_{mg}</td>
<td>14</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Proposition 1. The lower bound on $L_{\text{max}}$ for a machine group is a non-increasing function of the number of workers assigned to the machine group.

Proof. If assigning $j$ workers to machine group $i$ yields $\text{LB}_{mg_i}(j)$, then adding $k - j > 0$ (possibly idle) workers must yield $\text{LB}_{mg_i}(k) \leq \text{LB}_{mg_i}(j)$.

Proposition 1 implies that the addition of one or more workers to a machine group may improve the lower bound on $L_{\text{max}}$ for the machine group, and at worst keeps it the same. The example also illustrates that the lower bound on $L_{\text{max}}$ for a machine group cannot be characterized as a discrete convex function of the number of workers assigned to the machine group, where a function $g(j)$ defined on the integers with first difference $\Delta g(j) = g(j + 1) - g(j)$ is said to be a “discrete convex function” if the second difference $\Delta^2 g(j) = g(j + 2) - 2g(j + 1) + g(j)$ satisfies $\Delta^2 g(j) \geq 0$ for all $j$. In particular, for the function $g(j) = \text{LB}_{mg}(j)$ in Table 2, we see that with $j = 5$ the second difference is

$$\Delta^2 \text{LB}_{mg}(5) = \text{LB}_{mg}(7) - 2 \cdot \text{LB}_{mg}(6) + \text{LB}_{mg}(5)$$

$$= 2 - 2 \cdot 3 + 3 = -1 \not\geq 0.$$

Because the lower bound on $L_{\text{max}}$ for a machine group is not a discrete convex function of the number of workers assigned to the machine group, there is no simple search for an allocation $\theta^*$ yielding the smallest $\text{LB}_\theta$ value. However, a relatively straightforward search for such an allocation is formulated in the next section.

4.3. Finding an allocation with the smallest $\text{LB}_\theta$ value for the DRC job shop

4.3.1. An initial allocation

The following search procedure requires an initial allocation of workers to machine groups in the job shop. Each machine group must have at least one worker assigned to it; thus, there are only $W - S$ workers
that must be allocated. Let $\text{Proc}_i$ denote the sum of the processing times of all jobs that must be processed on the machines in machine group $i$, and let $\text{Proc}_{JS}$ denote the sum of the processing times of all the jobs that must be processed in the job shop. Then the additional number of workers that will be allocated to machine group $i$ is

$$\max \left( \min \left\{ \left( \frac{\text{Proc}_i}{\text{Proc}_{JS}} \right) (W - S), \lfloor \frac{|M_i|}{1} \rfloor - 1, 0 \right\} \right). \quad (5)$$

Due to rounding down and the maximum worker constraint, there may be workers that still need allocation after the $S$ machine groups have been allocated workers using Equation (5). In this case, the remaining workers are assigned sequentially to the machine groups with the largest values of the remainder

$$\left\{ \left( \frac{\text{Proc}_i}{\text{Proc}_{JS}} \right) (W - S) \right\} - \left\lfloor \left( \frac{\text{Proc}_i}{\text{Proc}_{JS}} \right) (W - S) \right\rfloor$$

that do not already have $|M_i|$ workers assigned to them, where $i = 1, \ldots, S$.

4.3.2. The Lower Bound Search Algorithm (LBSA)

Define a constraining machine group of allocation $\theta$ to be a machine group $i$ that has the largest value of $\text{LB}_{mg_i}(\theta_{w_i})$ for $i = 1, \ldots, S$. Note that in general allocation $\theta$ may have multiple constraining machine groups, each yielding the same (maximal) value of the lower bound on $L_{\text{max}}$ for that machine group.

The following observations can be made regarding how to find an allocation of workers to machine groups yielding the smallest $\text{LB}_{\theta}$ value for the DRC job shop:

(i) Given an allocation $\theta$, let machine group $k$ be a constraining machine group, where

$$K(\theta) = \arg \max_{1 \leq l \leq S} \{\text{LB}_{mg_l}(\theta_{w_l})\}$$

denotes the index-set of all constraining machine groups for allocation $\theta$ and $k \in K(\theta)$. If $w_k(\theta) = |M_k|$ for any $k \in K(\theta)$, then $\theta$ is an allocation yielding the smallest $\text{LB}_{\theta}$ value for the DRC job shop: a machine group with the largest $L_{\text{max}}$ value has one worker for every machine, and since the lower bound for the job shop is computed as the largest machine group $L_{\text{max}}$ value, this is the best lower bound that can be achieved.

(ii) Given an allocation $\theta$, a machine group $j$ cannot or will not have a worker unassigned from it if $w_j(\theta) = 1$, i.e., there is only one worker assigned to it presently. On the other hand, if $2 \leq w_j(\theta) \leq |M_j|$, then unassigning a worker from machine group $j$ and reassigning that worker to machine group $k$ leads to one of the two following cases:

CASE 1: If

$$\text{LB}_{mg_j}(\theta_{w_j}(\theta) - 1) \leq \text{LB}_{\theta},$$

then this new allocation may be better than allocation $\theta$. At this point a new allocation $\theta'$ can be specified: $\theta'$ is the same as $\theta$ except machine group $k$ has one more worker and machine group $j$ has one less.

CASE 2: If

$$\text{LB}_{mg_j}(\theta_{w_j}(\theta) - 1) > \text{LB}_{\theta},$$

then this new allocation is not better than the previous one. The index $j$ can be added to the set $F(\theta)$, where

$$F(\theta) = \{l : w_l(\theta) = 1 \text{ or } \text{LB}_{mg_l}(w_l(\theta) - 1) > \text{LB}_{\theta} \text{ for } 1 \leq l \leq S\}.$$

There are still $|\{1, \ldots, S\} \setminus F(\theta)|$ different machine groups that can have a worker unassigned from them in the attempt to find a better allocation.
(iii) Define a neighbor of an allocation \( \vartheta \) with respect to a machine group \( \ell \) (for \( 1 \leq \ell \leq S \)) to be an allocation \( \Omega \) such that if
\[
\vartheta = [w_1(\vartheta), \ldots, w_S(\vartheta)],
\]
then for some \( j \) satisfying \( 1 \leq j \leq S, j \neq \ell \), and \( w_j(\vartheta) \geq 2 \), we have
\[
\Omega = \vartheta - e_j + e_\ell,
\]
where for \( u = 1, \ldots, S \), the \( S \)-dimensional row vector \( e_u \) has its \( u \)th element equal to 1 and all other elements are equal to 0.

Given an allocation \( \vartheta \), suppose that \( \ell = k \in K(\vartheta) \). Since there are \( S \) machine groups in the job shop, an allocation \( \vartheta \) has up to \( S - 1 \) neighboring allocations with respect to the selected constraining machine group \( k \). To show that allocation \( \vartheta \) yields the smallest LB\( \vartheta \) value for the DRC job shop, we must show that for each of the neighbors of allocation \( \vartheta \) with respect to the selected constraining machine group \( k \), the associated lower bound on \( L_{\text{max}} \) values are greater than or equal to LB\( \vartheta \). This is a direct consequence of the following consideration: any sequence of worker reassignments between pairs of machine groups that yields a net increase of 1 worker at the “receiving” machine group \( k \) is ultimately traced to a net decrease of 1 worker at some “donating” machine group \( j \), where \( j \neq k \); and this reassignment is uniquely associated with one of the allocations that are neighbors of allocation \( \vartheta \) with respect to machine group \( k \).

These observations motivate the Lower Bound Search Algorithm (LBSA, or Algorithm 2) which guarantees that, upon termination, an allocation yielding the smallest LB\( \vartheta \) value for the DRC job shop has been obtained. Note that \( w_i(\vartheta^\lambda) \) denotes the number of workers assigned to machine group \( i \) under allocation \( \vartheta^\lambda \) for iteration \( \lambda = 0, 1, \ldots \). The algorithm will terminate in a finite number of iterations since there are a finite number of allocations of the \( W \) workers to the \( S \) machine groups in the DRC job shop.

**Proposition 2.** The LBSA terminates at an allocation \( \vartheta^* \) satisfying Equation (4), i.e., \( \text{LB}_{\vartheta^*} \leq \text{LB}_{\vartheta} \) for every feasible allocation \( \vartheta \).

**Proof.** The LBSA terminates at allocation \( \vartheta^* \) if one of the following conditions is satisfied:

**CONDITION 1:** The number of workers \( w_\ell(\vartheta^*) \) assigned to any constraining machine group \( \ell \in K(\vartheta^*) \) is equal to the number of machines \( |M_\ell| \) in that machine group. No additional workers can be assigned to such a machine group.

**CONDITION 2:** It is not possible to reassign a worker to the currently selected constraining machine group \( k \) from any of the \( S - 1 \) other machine groups. By Step 2 of LBSA, this means that it is not possible to unassign a worker from any other machine group \( j \neq k \) because one of the following conditions is satisfied:

(a) The machine group \( j \) only has one worker assigned to it currently — that is, \( w_j(\vartheta^*) = 1 \);

(b) The lower bound on \( L_{\text{max}} \) for the machine group \( j \) after unassignment of one worker from that machine group will be greater than the current lower bound on \( L_{\text{max}} \) for the currently selected constraining machine group \( k \) — that is,
\[
\text{LB}_{mg_j} (w_j(\vartheta^*) - 1) > \text{LB}_{mg_k} (w_k(\vartheta^*)) = \text{LB}_{\vartheta^*};
\]
or

(c) The “donating” machine group \( j \) is already a constraining machine group (\( j \in K(\vartheta^*) \)) so that its lower bound on \( L_{\text{max}} \) is already equal to LB\( \vartheta^* \).

Assume that there exists another allocation \( \Omega \neq \vartheta^* \) such that
\[
\text{LB}_{\Omega} < \text{LB}_{\vartheta^*}.
\]
Algorithm 2 The Lower Bound Search Algorithm (LBSA).

$\lambda \leftarrow 0$.

Determine an initial allocation $\theta^0$ according to Section 4.3.1.

1. $\text{LB}_{\theta_{\lambda}} \leftarrow \max_{1 \leq l \leq S} \left\{ \text{LB}_{\text{mg}}(w_l(\theta_{\lambda})) \right\}$
   
   $K(\theta_{\lambda}) \leftarrow \arg \max_{1 \leq l \leq S} \left\{ \text{LB}_{\text{mg}}(w_l(\theta_{\lambda})) \right\}$
   
   If $w_k(\theta_{\lambda}) = |N_k|$ for any $k \in K(\theta_{\lambda})$
   
   STOP. RETURN ALLOCATION $\theta_{\lambda}$.  
   
   Else
   
   Choose $k \in K(\theta_{\lambda})$.
   
   $F(\theta_{\lambda}) \leftarrow K(\theta_{\lambda})$
   
   For $z \in \{1, \ldots, S\} \setminus F(\theta_{\lambda})$
   
   If $w_z(\theta_{\lambda}) = 1$
   
   $F(\theta_{\lambda}) \leftarrow F(\theta_{\lambda}) \cup \{z\}$

2. If $\{1, \ldots, S\} \setminus F(\theta_{\lambda}) \neq \emptyset$
   
   $J(\theta_{\lambda}) \leftarrow \arg \min_{1 \leq l \leq S, l \notin F(\theta_{\lambda})} \left\{ \text{LB}_{\text{mg}}(w_l(\theta_{\lambda})) \right\}$
   
   Choose $j \in J(\theta_{\lambda})$.
   
   If $\text{LB}_{\text{mg}}(w_j(\theta_{\lambda}) - 1) > \text{LB}_{\theta_{\lambda}}$
   
   $F(\theta_{\lambda}) \leftarrow F(\theta_{\lambda}) \cup \{j\}$
   
   Go to 2.

   Else
   
   $\theta_{\lambda+1} \leftarrow \theta_{\lambda}$
   
   $w_j(\theta_{\lambda+1}) \leftarrow w_j(\theta_{\lambda}) - 1$
   
   $w_k(\theta_{\lambda+1}) \leftarrow w_k(\theta_{\lambda}) + 1$
   
   If $\theta_{\lambda+1} \in \{\theta^0, \ldots, \theta_{\lambda}\}$
   
   $F(\theta_{\lambda}) \leftarrow F(\theta_{\lambda}) \cup \{j\}$
   
   Go to 2.

   Else
   
   $\lambda \leftarrow \lambda + 1$
   
   Go to 1.

   Else
   
   STOP. RETURN ALLOCATION $\theta_{\lambda}$. 

14
Then by the definition of a constraining machine group, for any constraining machine group $\ell \in K(\vartheta^*)$ we must have

$$LB_{mg_\ell}(w_\ell(\Omega)) \leq LB_\Omega \leq LB_{\vartheta^*} = LB_{mg_\ell}(w_\ell(\vartheta^*)).$$

By Proposition 1, Equation (7) implies that

$$w_\ell(\Omega) > w_\ell(\vartheta^*)$$

for every $\ell \in K(\vartheta^*)$. From Equation (8) it follows that $w_\ell(\vartheta^*) < |M_\ell|$ for each constraining machine group $\ell \in K(\vartheta^*)$.

If LBSA terminated because CONDITION 1 was satisfied with the currently selected constraining machine group $k$, then $w_k(\vartheta^*) = |M_k|$, which contradicts Equation (8). Thus under the assumption of Equation (6), LBSA must have terminated because CONDITION 2 was satisfied. In this situation, Equation (8) implies that to move from allocation $\vartheta^*$ to allocation $\Omega$, we must assign $w_k(\Omega) - w_k(\vartheta^*) > 0$ additional workers to the currently selected constraining machine group $k$; and this must be accomplished by unassigning $w_k(\Omega) - w_k(\vartheta^*)$ workers from some of the other $S - 1$ machine groups. Because LBSA terminated at allocation $\vartheta^*$ with CONDITION 2 being satisfied, we see that all machine groups $j \neq k$ must satisfy one of conditions (a), (b), or (c). However, any machine group satisfying condition (a) above cannot have even one worker unassigned from it. Moreover, for any machine group $j$ satisfying condition (b), we have

$$LB_{mg_j}(w_j(\vartheta^*) - u) \geq LB_{mg_j}(w_j(\vartheta^*) - 1) > LB_{\vartheta^*}$$

for $u = 1, 2, \ldots$; and thus the move from allocation $\vartheta^*$ to allocation $\Omega$ cannot involve unassigning $u$ workers ($u \geq 1$) from any machine group $j$ satisfying condition (b) because otherwise we would have

$$LB_\Omega \geq LB_{mg_j}(w_j(\vartheta^*) - u) > LB_{\vartheta^*},$$

contradicting Equation (6). The only remaining machine groups from which workers can be unassigned must satisfy condition (c) — that is, the set $K(\vartheta^*)$ of constraining machine groups for allocation $\vartheta^*$. But allocation $\Omega$ cannot be obtained from allocation $\vartheta^*$ by unassigning $u$ workers ($u \geq 1$) from any machine group $\ell \in K(\vartheta^*)\setminus\{k\}$, because otherwise we would have

$$LB_\Omega \geq LB_{mg_\ell}(w_\ell(\vartheta^*) - u) \geq LB_{mg_\ell}(w_\ell(\vartheta^*)) = LB_{\vartheta^*}$$

for $\ell \in K(\vartheta^*)\setminus\{k\}$ and $u = 1, 2, \ldots$.

Again contradicting Equation (6).

Combining the conclusions of the last paragraph with the observation that LBSA must terminate at some allocation $\vartheta^*$ in a finite number of steps with either CONDITION 1 or CONDITION 2 being satisfied, we see that there cannot exist an allocation $\Omega \neq \vartheta^*$ satisfying Equation (6). Therefore LBSA's terminating allocation $\vartheta^*$ must achieve the minimum (smallest) $LB_{\vartheta}$ value for the DRC job shop. \hfill \square

Proposition 2 implies that if there is a unique allocation yielding the smallest $LB_{\vartheta}$ value, then LBSA will find it. On the other hand, if there are multiple such allocations, then LBSA will find one of them.
5. Experimentation

Due to the computational complexity of this problem a heuristic scheduler is used to evaluate different allocations. Although any heuristic scheduler would work, the Virtual Factory was chosen because of its proven track record in generating good schedules for job shop scheduling problems in which minimization of $L_{\text{max}}$ was the objective (see [20, 26, 27, 28, 29, 30, 31]). Given an allocation $\theta$ of workers to machine groups, we let $\text{VF}_{\theta}$ denote the value of $L_{\text{max}}$ for the schedule delivered by the Virtual Factory using allocation $\theta$. Note that the schedule generated by the Virtual Factory is a feasible working schedule, thus $\text{VF}_{\theta}$ is an actual maximum lateness ($L_{\text{max}}$) value for the problem being considered given the allocation $\theta$. The allocations in the set

$$A^{\text{VFB}} = \arg \min \{ \text{VF}_{\theta} : \theta \text{ is feasible} \}$$

are called VF–best allocations because any allocation $\theta^{\text{VFB}} \in A^{\text{VFB}}$ yields

$$\text{VF}_{\theta^{\text{VFB}}} = \min \{ \text{VF}_{\theta} : \theta \text{ is feasible} \},$$

the smallest value of $L_{\text{max}}$ for the current problem that can be attained by the Virtual Factory. It is of course possible to find a VF–best allocation by direct enumeration of all feasible allocations; but in general this approach is computationally prohibitive. For each problem instance we have $\text{VF}_{\theta^{*}} \geq \text{VF}_{\theta^{\text{VFB}}}$. The difference $\text{VF}_{\theta} - \text{LB}_{\theta^{*}}$ indicates how well an allocation $\theta$ performs relative to LB$_{\theta^{*}}$. The quantity $\text{VF}_{\theta^{*}} - \text{VF}_{\theta^{\text{VFB}}}$ measures the difference in performance between allocation $\theta^{*}$ and allocation $\theta^{\text{VFB}}$. If the difference $\text{VF}_{\theta^{*}} - \text{VF}_{\theta^{\text{VFB}}} = 0$, then $\theta^{*}$ is also a VF–best allocation. Let $\theta^{0}$ denote LBSA's initial allocation as detailed in Section 4.3.1. The average difference $\text{VF}_{\theta^{0}} - \text{LB}_{\theta^{*}}$ is depicted in the graphs that follow to provide a baseline for visually assessing whether the average value of the difference $\text{VF}_{\theta^{VFB}} - \text{LB}_{\theta^{*}}$ and average value of the difference $\text{VF}_{\theta^{*}} - \text{LB}_{\theta^{*}}$ are due to the problem nature or the solution approach.

If $\text{VF}_{\theta^{\text{VFB}}} - \text{LB}_{\theta^{*}} = 0$, then allocation $\theta^{\text{VFB}}$ is an optimal solution to the $J_{M}|W|L_{\text{max}}$ DRC job shop scheduling problem. On the other hand, if $\text{VF}_{\theta^{\text{VFB}}} - \text{LB}_{\theta^{*}} \neq 0$, then $\text{VF}_{\theta^{\text{VFB}}}$ provides an upper bound on $L_{\text{max}}$ for the problem; and $\text{VF}_{\theta^{\text{VFB}}} - \text{LB}_{\theta^{*}}$ is an upper bound on the largest possible improvement that can be achieved using other solution methods.

5.1. Experimental design

An experiment was performed to explore the performance of allocations $\theta^{*}$ and $\theta^{\text{VFB}}$ for different types of DRC job shops, with different staffing levels, and over a variety of due-date ranges. The job shop used in our experiments was modeled on the furniture manufacturing facility that prompted this research problem, and has 80 machines split into 10 machine groups, where each machine group has 8 machines. There are 1,200 jobs that require processing. The due-date of each job is a function of the due-date range, which varies from 200 to 3,000 in increments of 400. The upper bound of 3,000 was chosen following experiments showing that, when maximum due-date exceeded 3,000, the maximum flow time in the job shop ($F_{\text{max}}$) was significantly less than the latest due-date, an unrealistic scenario. The processing time of each operation is a discrete random variable drawn from a Uniform(1, 40) distribution. Every job has between 6 and 10 operations, and at most 3 of these operations can occur in the same machine group (not necessarily consecutively). When generating benchmark problems for job shops in which $L_{\text{max}}$ is the objective to be minimized, Demirkol et al. [36] use the parameters of due-date range and the expected number of tardy jobs to determine job due-dates. However, Hodgson et al. [26] find that if the due-date “range is held constant, the tightness of the due-dates does not affect the optimal sequence”. Thus even though the expected number of tardy jobs (and the number of jobs) is not varied, by varying the due-date range we in fact vary the difficulty of problems being solved. Four different overall job shop staffing levels are considered, namely, 60%, 70%, 80%, and 90% (a 60% staffing level corresponds to there being 48 workers available for allocation to the 80 machines).
The type of job shop, which is reflected in the symmetric (balanced) and asymmetric (unbalanced) loading of the machine groups, also forms part of the experimental design. A description of the symmetric job shop can be found in Section 5.2.1, while that of the asymmetric job shop can be found in Section 5.2.2.

The combination of job shop type, staffing level, and the due-date range of the jobs defines a class of DRC job shop scheduling problems. For each class of problems, 200 randomly generated problem instances were solved. The $i$th randomly generated problem instance for a given problem type (job shop type and due-date range combination) is the same problem seen by all staffing levels for $i = 1, \ldots, 200$ (see Appendix A for further details), thus the performance comparisons across staffing levels were sharper than we would have achieved by randomly generating a new set of 200 problems for each staffing level; see, for example, Section 5 of Chang et al. [37].

All experimentation was performed on computers that had an Intel Xeon W3520 processor running at 2.67GHz with 4GB of RAM.

Experimental results regarding the sensitivity of the lower bound to some of the experimental job shop assumptions can be found in Appendix C.

5.2. Evaluation of allocation $\hat{\theta}^*$

The graphs presented in Section 5.2.1 and 5.2.2 show the average performance of allocation $\hat{\theta}^0$, allocation $\hat{\theta}^*$, and allocation $\hat{\theta}^{\text{VFB}}$ relative to the minimum lower bound on $L_{\text{max}}$ delivered by LBSA (i.e., $\text{LB}_{\hat{\theta}^*}$). Each point on the piecewise linear curve (polyline) labeled “Allocation $\hat{\theta}^0$” is the average of 200 differences $\text{VF}_{\hat{\theta}^0} - \text{LB}_{\hat{\theta}^*}$ for 200 independently generated instances of that particular problem type. Each point on the polyline labeled “Allocation $\hat{\theta}^*$” is the average of 200 differences $\text{VF}_{\hat{\theta}^*} - \text{LB}_{\hat{\theta}^*}$ for the same 200 independently generated instances of that particular problem type. Similarly, each point on the polyline labeled “Allocation $\hat{\theta}^{\text{VFB}}$” is the average of 200 differences $\text{VF}_{\hat{\theta}^{\text{VFB}}} - \text{LB}_{\hat{\theta}^*}$ for the same 200 independently generated instances of that particular problem type.

5.2.1. Symmetric DRC job shop

In the symmetric job shop, because every machine group has an equal probability of being on a job’s route, on average each of the 10 machine groups has 10% of the job shop workload pass through it. The graphs pertaining to the symmetric job shop can be seen in Figure 3. Table 3 shows the average $\text{LB}_{\hat{\theta}^*}$ values (in units of average operation processing time) for the four different staffing levels across the entire due-date range in the symmetric job shop, along with the linear fit to the 1,600 $\text{LB}_{\hat{\theta}^*}$ values and the corresponding value of $R^2$, the coefficient of determination for the fit. Table 4 shows the same information, but for $\text{VF}_{\hat{\theta}^{\text{VFB}}}$. In the linear fit $y = mx + b$, the variable $y$ represents the value of $\text{LB}_{\hat{\theta}^*}$, $\text{VF}_{\hat{\theta}^*}$, or $\text{VF}_{\hat{\theta}^{\text{VFB}}}$ depending on the fit, that can be expected from a randomly generated problem instance when the due-date range is $x$.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing</th>
<th>70% Staffing</th>
<th>80% Staffing</th>
<th>90% Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>222.0</td>
<td>179.9</td>
<td>151.5</td>
<td>141.9</td>
</tr>
<tr>
<td>600</td>
<td>202.9</td>
<td>161.0</td>
<td>132.7</td>
<td>123.3</td>
</tr>
<tr>
<td>1000</td>
<td>183.8</td>
<td>142.0</td>
<td>113.9</td>
<td>104.6</td>
</tr>
<tr>
<td>1400</td>
<td>164.7</td>
<td>123.0</td>
<td>95.0</td>
<td>85.8</td>
</tr>
<tr>
<td>1800</td>
<td>145.6</td>
<td>104.0</td>
<td>76.1</td>
<td>67.1</td>
</tr>
<tr>
<td>2200</td>
<td>126.5</td>
<td>85.0</td>
<td>57.2</td>
<td>48.8</td>
</tr>
<tr>
<td>2600</td>
<td>107.4</td>
<td>66.0</td>
<td>38.6</td>
<td>31.9</td>
</tr>
<tr>
<td>3000</td>
<td>88.3</td>
<td>47.1</td>
<td>21.6</td>
<td>19.7</td>
</tr>
</tbody>
</table>

Linear Fit, $y = -0.0478x + 231.6 \quad -0.0475x + 189.5 \quad -0.0467x + 160.5 \quad -0.0446x + 149.2$

$R^2 = 0.9939 \quad 0.9976 \quad 0.9975 \quad 0.9808$
Figure 3: Average deviation from the smallest $LB_0$ value, symmetric job shop.
Table 4: Average VF_{VFB} value (in units of average operation processing time) together with the associated linear fit for the four different staffing levels in the symmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing</th>
<th>70% Staffing</th>
<th>80% Staffing</th>
<th>90% Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>223.0</td>
<td>181.4</td>
<td>154.0</td>
<td>142.3</td>
</tr>
<tr>
<td>600</td>
<td>203.9</td>
<td>162.4</td>
<td>135.4</td>
<td>123.9</td>
</tr>
<tr>
<td>1000</td>
<td>184.7</td>
<td>143.4</td>
<td>116.9</td>
<td>105.3</td>
</tr>
<tr>
<td>1400</td>
<td>165.6</td>
<td>124.6</td>
<td>98.4</td>
<td>86.6</td>
</tr>
<tr>
<td>1800</td>
<td>146.5</td>
<td>105.8</td>
<td>80.0</td>
<td>68.1</td>
</tr>
<tr>
<td>2200</td>
<td>127.4</td>
<td>87.0</td>
<td>61.5</td>
<td>50.1</td>
</tr>
<tr>
<td>2600</td>
<td>108.4</td>
<td>68.4</td>
<td>43.5</td>
<td>33.5</td>
</tr>
<tr>
<td>3000</td>
<td>89.5</td>
<td>49.9</td>
<td>26.9</td>
<td>21.4</td>
</tr>
</tbody>
</table>

Linear Fit, \( y = -0.0477x + 232.5 \)  
\( R^2 \) 0.9940 0.9975 0.9980 0.9848

The following observations can be made regarding the experimental results in the symmetric job shop:

**Observation 1:** In all but the 60% staffing case, the average performance relative to LB_{\theta^*} of both allocation \( \theta^* \) and allocation \( \theta^{VFB} \) was worse as the due-date range increased. This trend was most apparent in the 80% staffing case.

**Observation 2:** For all four staffing levels, the average difference \( VF_{\theta^*} - VF_{\theta^{VFB}} = (VF_{\theta^*} - LB_{\theta^*}) - (VF_{\theta^{VFB}} - LB_{\theta^*}) \) stayed relatively constant across the due-date range. The \( VF_{\theta^*} \) value was, on average, no more than a single expected operation processing time greater than the \( VF_{\theta^{VFB}} \) value, and in the case of 60% staffing, the two values were nearly equivalent.

**Observation 3:** The “Linear Fit” and “\( R^2 \)” results in Table 4 indicated that both the individual and average values of \( VF_{\theta^{VFB}} \) decreased linearly as the due-date range increased; Observation 2 implied that the average \( VF_{\theta^*} \) value also decreased linearly as the due-date range increased (a result confirmed by the linear fits shown in Table B.12 of Appendix B).

**Observation 4:** In the cases of 70% and 80% staffing, there is a statistically significant difference (at the \( \alpha = 0.05 \) level) between the magnitudes of the slopes of the linear fits to the \( LB_{\theta^*} \) values (in Table 3) and the magnitudes of the slopes of the corresponding linear fits to the \( VF_{\theta^{VFB}} \) values (in Table 4).

From Observations 2, 3, and 4, it is not clear whether Observation 1 was solely because (a) the quality of the schedule being produced by the Virtual Factory deteriorated as the due-date range increased, or (b) the quality of the lower bound deteriorated as the due-date increased. However, it is likely that Observation 1 is due to a combination of (a) and (b).

We also observed that:

**Observation 5:** The average difference \( VF_{\theta^0} - VF_{\theta^*} \) stayed relatively constant (and almost 0) across the due-date range.

In light of Observation 2 and Observation 5, we concluded that the stability of the average difference \( VF_{\theta^*} - VF_{\theta^{VFB}} \) was most likely due to the problem nature, and not the solution approach; after all, allocation \( \theta^0 \) was determined using a completely different methodology than either allocation \( \theta^* \) or allocation \( \theta^{VFB} \).

Observation 5 may raise questions about the benefits of exploiting the allocation \( \theta^* \) delivered by LBSA. However, LBSA provides a lower bound on \( L_{\text{max}} \) that enables the optimality of a given allocation \( \theta \) to be established if \( VF_{\theta} = LB_{\theta^*} \). In addition, if \( VF_{\theta} \neq LB_{\theta^*} \), then the difference \( VF_{\theta} - LB_{\theta^*} \) is an upper bound on the optimality gap. Moreover, in the next section and in Lobo et al. [13], we see that (a) generally the difference \( VF_{\theta^0} - VF_{\theta^*} \) can be substantial, and (b) the allocation \( \theta^* \) delivered by LBSA can be critical to the identification of optimal or near-optimal allocations and the associated schedules.
5.2.2. Asymmetric DRC job shop

Asymmetry in the job shop is created by altering the probability that a machine group is on a job’s route. Without loss of generality, machine groups 1 through 4 were chosen to be the machine groups with a greater probability of having a job routed through them (see Table 5). For example, machine group 2 had 14% of the workload pass through it, whereas machine group 6 had just 8% of the workload pass though it. The graphs pertaining to the asymmetric job shop can be seen in Figure 4. Table 6 shows the average LB\(_{d^*}\) values (in units of average operation processing time) for the four different staffing levels across the entire due-date range in the asymmetric job shop, along with the linear fit to the 1,600 LB\(_{d^*}\) values and the corresponding value of \(R^2\), the coefficient of determination for the fit. Table 7 shows the same information, but for VF\(_{\hat{d}^{VFB}}\).

<table>
<thead>
<tr>
<th>Machine Group</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.14</td>
</tr>
<tr>
<td>2</td>
<td>0.14</td>
</tr>
<tr>
<td>3</td>
<td>0.14</td>
</tr>
<tr>
<td>4</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.08</td>
</tr>
<tr>
<td>6</td>
<td>0.08</td>
</tr>
<tr>
<td>7</td>
<td>0.08</td>
</tr>
<tr>
<td>8</td>
<td>0.08</td>
</tr>
<tr>
<td>9</td>
<td>0.08</td>
</tr>
<tr>
<td>10</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 6: Average LB\(_{d^*}\) value (in units of average operation processing time) together with the associated linear fit for the four different staffing levels in the asymmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing</th>
<th>70% Staffing</th>
<th>80% Staffing</th>
<th>90% Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>211.8</td>
<td>186.6</td>
<td>185.5</td>
<td>185.5</td>
</tr>
<tr>
<td>600</td>
<td>192.8</td>
<td>167.8</td>
<td>166.8</td>
<td>166.8</td>
</tr>
<tr>
<td>1000</td>
<td>173.7</td>
<td>148.9</td>
<td>147.9</td>
<td>147.9</td>
</tr>
<tr>
<td>1400</td>
<td>154.7</td>
<td>129.9</td>
<td>129.0</td>
<td>129.0</td>
</tr>
<tr>
<td>1800</td>
<td>135.6</td>
<td>111.0</td>
<td>110.1</td>
<td>110.1</td>
</tr>
<tr>
<td>2200</td>
<td>116.5</td>
<td>92.1</td>
<td>91.2</td>
<td>91.2</td>
</tr>
<tr>
<td>2600</td>
<td>97.5</td>
<td>73.3</td>
<td>72.6</td>
<td>72.6</td>
</tr>
<tr>
<td>3000</td>
<td>78.4</td>
<td>55.0</td>
<td>54.3</td>
<td>54.3</td>
</tr>
</tbody>
</table>

Linear Fit, \(y = -0.0476x + 221.3\) 
\(R^2 = 0.9921\)

Table 7: Average VF\(_{\hat{d}^{VFB}}\) value (in units of average operation processing time) together with the associated linear fit for the four different staffing levels in the asymmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing</th>
<th>70% Staffing</th>
<th>80% Staffing</th>
<th>90% Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>214.1</td>
<td>187.5</td>
<td>185.5</td>
<td>185.5</td>
</tr>
<tr>
<td>600</td>
<td>195.1</td>
<td>168.7</td>
<td>166.8</td>
<td>166.8</td>
</tr>
<tr>
<td>1000</td>
<td>176.1</td>
<td>149.9</td>
<td>147.9</td>
<td>147.9</td>
</tr>
<tr>
<td>1400</td>
<td>157.3</td>
<td>131.1</td>
<td>129.0</td>
<td>129.0</td>
</tr>
<tr>
<td>1800</td>
<td>138.5</td>
<td>112.1</td>
<td>110.1</td>
<td>110.1</td>
</tr>
<tr>
<td>2200</td>
<td>119.7</td>
<td>93.4</td>
<td>91.2</td>
<td>91.2</td>
</tr>
<tr>
<td>2600</td>
<td>100.8</td>
<td>74.8</td>
<td>72.6</td>
<td>72.6</td>
</tr>
<tr>
<td>3000</td>
<td>82.1</td>
<td>56.5</td>
<td>54.4</td>
<td>54.4</td>
</tr>
</tbody>
</table>

Linear Fit, \(y = -0.0471x + 223.4\) 
\(R^2 = 0.9914\)

The following observations can be made regarding the experimental results in the asymmetric job shop: **Observation 6**: The average performance relative to LB\(_{d^*}\) of both allocation \(\hat{d}^*\) and allocation \(\hat{d}^{VFB}\) was worse as the due-date range increased. This trend was most apparent in the 60% staffing case, and almost negligible in the 80% and 90% staffing cases.
Figure 4: Average deviation from the smallest $\hat{V}_B$ value, asymmetric job shop.
Observation 7: For all four staffing levels, the average difference $\text{VF}_{\theta^*} - \text{VF}_{\text{VFB}}$ stayed relatively constant across the due-date range. The $\text{VF}_{\theta^*}$ value was, on average, no more than two expected operation processing times greater than the $\text{VF}_{\text{VFB}}$ value, and in the cases of 80% and 90% staffing, the two values were nearly equivalent.

Observation 8: The “Linear Fit” and “$R^2$” results in Table 7 indicated that both the individual and average values of $\text{VF}_{\text{VFB}}$ decreased linearly as the due-date range increased; Observation 7 implied that the average $\text{VF}_{\theta^*}$ value also decreased linearly as the due-date range increased (a result confirmed by the linear fits shown in Table B.11 of Appendix B).

Observation 9: In the case of 60% staffing, there is a statistically significant difference (at the $\alpha = 0.05$ level) between the magnitude of the slope of the linear fit to the $\text{LB}_{\theta^*}$ values (in Table 6) and the magnitude of the slope of the corresponding linear fit to the $\text{VF}_{\text{VFB}}$ values (in Table 7).

From Observations 7, 8, and 9, it is not clear whether Observation 6 was solely because (a) the quality of the schedule being produced by the Virtual Factory deteriorated as the due-date range increased, or (b) the quality of the lower bound deteriorated as the due-date increased. However, it is again likely that Observation 6 is due to a combination of (a) and (b).

In light of Observation 7 and the observation that the average difference $\text{VF}_{\theta^*} - \text{VF}_{\theta^*}$ stayed relatively constant across the entire due-date range, we again concluded that the stability of the average difference $\text{VF}_{\theta^*} - \text{VF}_{\text{VFB}}$ was most likely due to the problem nature, and not the solution approach. The value of LBSA-delivered allocation $\theta^*$ was most apparent in the 70% staffing case, where the average difference $\text{VF}_{\theta^*} - \text{VF}_{\theta^*}$ was typically two average operation processing times.

One possible reason that the 80% and 90% staffed job shops behaved so differently from the 60% and 70% staffed job shops could be that an asymmetric job shop with 80% staffing has enough workers available so that the machine groups through which the $L_{\text{max}}$ job is routed (the bottlenecks) already have the maximum number of workers allocated to them. This explanation is strengthened by examining how an increase in staffing level affected the average $\text{VF}_{\text{VFB}}$ value (the conclusions that follow can also be obtained from the results presented in Table B.11 of Appendix B, which displays the same information as Table 7, but for $\text{VF}_{\theta^*}$). The slopes of the linear fits in Table 7 are approximately equal, indicating that the linear fits are parallel to each other. Thus from the $y$-intercept of the linear fits, we can conclude that, across the entire due-date range there was on average a decrease of 26.7 average operation processing times in the average $\text{VF}_{\text{VFB}}$ value if the staffing level was increased from 60% to 70%. Similarly there was on average a decrease of 1.8 average operation processing times in the average value of $\text{VF}_{\text{VFB}}$ if the staffing level was increased from 70% to 80%; but there was virtually no decrease if the staffing level was increased from 80% to 90%. Thus the additional workers available for allocation at the 90% staffing level had no impact on the schedule’s $L_{\text{max}}$ value, indicating that in an asymmetric job shop a high level of staffing may not be necessary to achieve optimal performance.

5.2.3. Computation time

In order to explain the computation times presented next, the following description of the function $\text{LB}_{\theta}$ for allocations $\theta$ surrounding allocation $\theta^*$ is necessary. If LBSA terminated at allocation $\theta^*$ with the currently selected constraining machine group $k$ because CONDITION 1 was satisfied, then in general there are a large number of allocations $\theta$ that surround allocation $\theta^*$ for which $\text{LB}_{\theta} = \text{LB}_{\theta^*}$. Such allocations can be obtained by unassignment and reassignment of workers in machine groups other than machine group $k$, and the “response surface” formed by the $\text{LB}_{\theta}$ values of these allocations $\theta$ surrounding allocation $\theta^*$ is relatively flat in nature. On the other hand, if LBSA terminated because CONDITION 2 was satisfied, then the allocations that surround allocation $\theta^*$ satisfy $\text{LB}_{\theta} \geq \text{LB}_{\theta^*}$, and typically they satisfy $\text{LB}_{\theta} > \text{LB}_{\theta^*}$. Such allocations are the neighbors of allocation $\theta^*$, the neighbors of the neighbors of allocation $\theta^*$, and so on; the “response surface” has a bowl–type (relatively convex) shape to it, with $\text{LB}_{\theta^*}$ at the bottom of the
bowl. From Table 8 we see that with 60% and 70% staffing, LBSA terminated with CONDITION 2 satisfied in all problem instances. However, as the staffing level is increased to 80% and then 90%, the number of problem instances in which LBSA terminated with CONDITION 1 satisfied increased dramatically.

Table 8: Percent of replications in which LBSA terminated because CONDITION 1 was satisfied, symmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing</th>
<th>70% Staffing</th>
<th>80% Staffing</th>
<th>90% Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0.0</td>
<td>0.0</td>
<td>3.5</td>
<td>97.5</td>
</tr>
<tr>
<td>600</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>97.0</td>
</tr>
<tr>
<td>1000</td>
<td>0.0</td>
<td>0.0</td>
<td>4.5</td>
<td>98.0</td>
</tr>
<tr>
<td>1400</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>98.0</td>
</tr>
<tr>
<td>1800</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>99.0</td>
</tr>
<tr>
<td>2200</td>
<td>0.0</td>
<td>0.0</td>
<td>6.0</td>
<td>99.5</td>
</tr>
<tr>
<td>2600</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
<td>100.0</td>
</tr>
<tr>
<td>3000</td>
<td>0.0</td>
<td>0.0</td>
<td>27.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Tables 3 and 6 show that as the due-date range increased, the average $L_{\phi_\pi}$ value decreased linearly. One reason for this was that, as the due-date range increased, the processing of jobs with later due-dates could be increasingly delayed in favor of jobs with more critical due-dates, while the lower bound on $L_{\max}$ value was less adversely affected. Furthermore, if the number of jobs was kept constant while the due-date range increased, then the larger the due-date range the smaller the lower bound on $L_{\max}$ was, since an increasing number of jobs could be delayed increasingly in favor of the critical jobs. This also meant that the impact of having one less worker assigned to a bottleneck machine group became less important, and thus as the due-date range increased, the number of allocations $\phi$ with $L_{\phi}$ near $L_{\phi*}$ increased. Thus as the due-date range increased, the “response surface” went from a bowl–type shape with relatively steep sides to a bowl–type shape with relatively shallow sides, or even to a “response surface” that was relatively flat in nature. Table 8 shows that, as expected, the number of problem instances in which LBSA terminated with CONDITION 1 satisfied increased dramatically as the staffing level increased.

We observed the following regarding the initial allocation:

**Observation 10:** As staffing level increased, the initial allocation was more likely to be allocation $\phi^*$.  
**Observation 11:** As due-date range increased, the initial allocation was less likely to be allocation $\phi^*$, especially when the staffing level was low.

The less likely the initial allocation was to be allocation $\phi^*$, the greater the number of allocations that had to be searched before allocation $\phi^*$ was found, and vice versa.

Table 9 shows, for the symmetric job shop, the sample mean $\bar{\mu}$ and the sample standard deviation $\sigma$ of the execution time needed to find and evaluate allocation $\phi^*$. The trends in Table 9 can be explained as follows. The time taken by LBSA depended on (a) the value of the lower bound on $L_{\max}$ for each machine group ($L_{\phi \text{mg}}$); and (b) the number of allocations that were searched before allocation $\phi^*$ was identified. In general, the lower the staffing level was, the larger $L_{\phi \text{mg}}$ was, the longer Algorithm 1 took, and the larger the network for each machine group was. Therefore at lower staffing levels, Algorithm 1 required larger execution times, and the network flow formulation for each machine group required more time to set up and solve. Along with Observation 10, these considerations explain why LBSA’s mean execution time decreased as the staffing level increased. Observation 11 reveals the reason that, especially in the lower staffing levels, LBSA’s mean execution time increased as the due-date range increased.

Table 10 shows, for the symmetric job shop, the sample mean $\bar{\mu}$ and the sample standard deviation $\sigma$ of the execution time needed to enumerate the allocation search space once allocation $\phi^*$ had been found and evaluated (enumeration is necessary when $\text{VF}_{\phi^*} - L_{\phi^*} \neq 0$). From the average execution times, we concluded that as both the percent staffing and the due-date range increased, the execution time needed to
enumerate the allocation search space tended to increase (but was not always the case). These trends can be explained as follows. In the enumeration procedure, only those allocations $\vartheta$ satisfying the condition for enumeration, i.e., $LB_\vartheta < VF_\vartheta$, were evaluated. The number of allocations $\vartheta$ with $LB_\vartheta$ near $LB_\vartheta^*$ increased as both the staffing level and due-date range increased (see the first and second paragraph of this section respectively). The average enumeration time for 80% staffing tended to be greater than the average enumeration time for 90% staffing for the following reason. As can be seen from Figure 3, at the same due-date range value, the average difference $VF_\vartheta^* - LB_\vartheta^*$ was greater across the entire due-date range in the case of 80% staffing compared with the case of 90% staffing. This means that, in general, there were more allocations that satisfied the condition for enumeration when the staffing level was 80%.

For the 60% staffing case the average time needed to enumerate the allocation search space was comparable to the average time needed to find and evaluate allocation $\vartheta^*$, except when the due-date range was 200. Inspection of the relevant data revealed two problem instances (of the 200) with very large enumeration times, however it is not clear why these specific problem instances required so many many allocations to be searched. When these two problems were excluded from the data set the resulting average was 31.9 with a standard deviation of 77.2. When the staffing level was 70%, 80%, or 90%, the average time needed to enumerate the allocation search space was significantly greater than the average time needed to find and evaluate allocation $\vartheta^*$. Similar results were obtained for the asymmetric job shop; all of these results clearly reveal the need for a search strategy that finds allocation $\vartheta^{VFB}$ without enumerating the allocation search space.

Table 9: Sample mean $\bar{\mu}$ and standard deviation $\bar{\sigma}$ of the time taken (in seconds) for LBSA in the symmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing $\bar{\mu}$</th>
<th>60% Staffing $\bar{\sigma}$</th>
<th>70% Staffing $\bar{\mu}$</th>
<th>70% Staffing $\bar{\sigma}$</th>
<th>80% Staffing $\bar{\mu}$</th>
<th>80% Staffing $\bar{\sigma}$</th>
<th>90% Staffing $\bar{\mu}$</th>
<th>90% Staffing $\bar{\sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>27.4 0.7</td>
<td>19.9 0.6</td>
<td>13.2 1.5</td>
<td>3.5 1.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>28.5 0.9</td>
<td>21.3 1.0</td>
<td>14.2 1.8</td>
<td>4.0 2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>29.7 1.2</td>
<td>22.6 1.2</td>
<td>15.1 2.0</td>
<td>4.2 2.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td>30.5 1.5</td>
<td>23.5 1.4</td>
<td>15.7 2.3</td>
<td>4.4 2.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td>31.8 1.9</td>
<td>24.6 1.8</td>
<td>16.5 2.6</td>
<td>4.5 2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2200</td>
<td>33.4 2.5</td>
<td>26.4 2.4</td>
<td>18.2 3.1</td>
<td>4.9 2.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2600</td>
<td>35.4 3.2</td>
<td>28.9 3.5</td>
<td>20.9 4.1</td>
<td>5.1 3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>39.3 5.4</td>
<td>37.0 7.6</td>
<td>27.8 9.2</td>
<td>3.8 2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Sample mean $\bar{\mu}$ and standard deviation $\bar{\sigma}$ of the time taken (in seconds) for the enumeration procedure in the symmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing $\bar{\mu}$</th>
<th>60% Staffing $\bar{\sigma}$</th>
<th>70% Staffing $\bar{\mu}$</th>
<th>70% Staffing $\bar{\sigma}$</th>
<th>80% Staffing $\bar{\mu}$</th>
<th>80% Staffing $\bar{\sigma}$</th>
<th>90% Staffing $\bar{\mu}$</th>
<th>90% Staffing $\bar{\sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>92.3 607.1</td>
<td>152.2 484.0</td>
<td>234.8 395.2</td>
<td>56.7 138.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>28.8 77.0</td>
<td>102.5 257.9</td>
<td>273.9 413.0</td>
<td>135.7 760.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>23.4 51.8</td>
<td>103.6 242.5</td>
<td>391.0 562.4</td>
<td>238.9 1166.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td>26.8 66.1</td>
<td>107.5 245.0</td>
<td>462.8 650.2</td>
<td>167.3 279.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td>21.9 45.9</td>
<td>131.6 305.5</td>
<td>568.1 715.0</td>
<td>208.5 440.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2200</td>
<td>20.0 41.7</td>
<td>179.9 376.6</td>
<td>663.1 790.2</td>
<td>525.8 960.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2600</td>
<td>21.9 41.1</td>
<td>269.1 590.5</td>
<td>808.2 828.0</td>
<td>1632.8 2843.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>34.7 79.1</td>
<td>418.7 960.4</td>
<td>1263.7 915.6</td>
<td>7877.8 5817.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the 60% staffing case the average time needed to enumerate the allocation search space was comparable to the average time needed to find and evaluate allocation $\vartheta^*$, except when the due-date range was 200. Inspection of the relevant data revealed two problem instances (of the 200) with very large enumeration times, however it is not clear why these specific problem instances required so much many allocations to be searched. When these two problems were excluded from the data set the resulting average was 31.9 with a standard deviation of 77.2. When the staffing level was 70%, 80%, or 90%, the average time needed to enumerate the allocation search space was significantly greater than the average time needed to find and evaluate allocation $\vartheta^*$. Similar results were obtained for the asymmetric job shop; all of these results clearly reveal the need for a search strategy that finds allocation $\vartheta^{VFB}$ without enumerating the allocation search space.
6. Conclusions and future research

In this article we address the problem of finding an allocation of workers to machine groups yielding the smallest LB value for the DRC job shop. Given an allocation of workers to machine groups, we first develop a method for obtaining the lower bound on \( L_{\max} \) for an individual machine group, and then take the maximum over all machine group lower bounds as the allocation lower bound. We then present a search algorithm to find an allocation \( \delta^* \) yielding the smallest LB value for the DRC job shop. The experimental results shown in Section 5.2 illustrate the following: (a) while allocation \( \delta^* \) is the allocation that yields the smallest LB value, it does not necessarily allow the schedule with the smallest \( L_{\max} \) value to be generated using the Virtual Factory; and (b) the lower bound on \( L_{\max} \) was not as effective as the due-date range increased, but the average difference between VF\( \delta^* \) and VF\( \delta^{VFB} \) was nearly constant across the entire due-date range for a given problem class. In addition to establishing a method to find a lower bound on \( L_{\max} \) for the problem at hand, the solution approach also establishes an upper bound on the problem: the value VF\( \delta^{VFB} \) is an upper bound for the \( J_{M1MW_{ij}} \text{DRC job shop scheduling problem.} \)

The experimental results presented show that, in general, there are allocations enabling the Virtual Factory to generate a schedule with a smaller \( L_{\max} \) value than VF\( \delta^* \). One way to obtain such an allocation is through enumeration, but the average execution times presented in Tables 9 and 10 indicate the impracticality of this solution method. Thus further research is needed to formulate heuristic search strategies that find allocation \( \delta^{VFB} \). In addition, the experimental results motivate the development of optimality criteria that can be used to verify when allocation \( \delta^* \) coincides with allocation \( \delta^{VFB} \), particularly when VF\( \delta^* \neq LB^* \). These issues are addressed in Lobo et al. [13], where we formulate criteria for verifying that the allocation yielding the smallest LB value actually minimizes \( L_{\max} \) for the DRC job shop; to handle situations in which these optimality criteria are not satisfied, we formulate and experimentally evaluate a heuristic search procedure for finding improved worker allocations. We believe that Lobo et al. [13], together with this article, provide the first method for allocating workers to machine groups in a DRC job shop so as to achieve a minimal or near-minimal value of \( L_{\max} \).

Appendix A. Randomly Generating a Problem Instance

Once the

1) number of jobs, machines, and machine groups,
2) minimum and maximum number of operations per job,
3) maximum number of operations that can occur in one machine group, and
4) distribution of the processing time of a single operation

have been specified, a randomly generated problem instance is uniquely identified by (a) the type of job shop (symmetric or asymmetric), (b) the due-date range, and (c) the replication number (which serves as an index to an array of seeds for the random number generator). In this way, for example, the \( j^{th} \) randomly generated problem instance of a symmetric job shop with a due-date range of 600 is the same problem instance seen by all four staffing levels.

In the experimental job shop considered in this article, without loss of generality, the job shop was divided into 10 machine groups so that \( M_1 = \{1, \ldots, 8\}, M_2 = \{9, \ldots, 16\}, \ldots, M_{10} = \{72, \ldots, 80\} \). For each job that passed through the job shop, the following were randomly generated:

(i) the due-date of the job,
(ii) the number of operations the job had,
(iii) the route of the job through the job shop (i.e., the ordered sequence of machines that the job visited),
and
the processing time of each operation.

A linear congruential generator [38] was used to generate random numbers that were uniformly distributed on the interval \((0, 1)\). The due-date of the job was a discrete random variable uniformly distributed between 0 and the due-date range. The number of operations a job had was a discrete random variable uniformly distributed between 6 and 10. For each operation, the machine group in which that operation occurred was a discrete random variable that was either (a) uniformly distributed between machine groups 1 through 10 if the job shop was symmetric; or (b) machine group \(i\) with the probability specified in Table 5 if the job shop was asymmetric. For a given operation, the machine on which that operation occurred was a discrete random variable uniformly distributed amongst the indices of the machines in that machine group. Since a job could have up to 3 operations in the same machine group (not necessarily consecutively), the job could also have up to 3 operations on the same machine. The processing time of each operation was a discrete random variable uniformly distributed between 1 and 40.

Appendix B. Linear Fits to Maximum Lateness Values

In the linear fit \(y = mx + b\), the variable \(y\) represents the value of \(VF_{\theta^*}\) that can be expected from a randomly generated problem instance when the due-date range is \(x\).

### Table B.11: Average \(VF_{\theta^*}\) value (in units of average operation processing time) together with the associated linear fit for the four different staffing levels in the asymmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing</th>
<th>70% Staffing</th>
<th>80% Staffing</th>
<th>90% Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>214.4</td>
<td>188.5</td>
<td>185.5</td>
<td>185.5</td>
</tr>
<tr>
<td>600</td>
<td>195.3</td>
<td>170.1</td>
<td>166.8</td>
<td>166.8</td>
</tr>
<tr>
<td>1000</td>
<td>176.3</td>
<td>151.5</td>
<td>147.9</td>
<td>147.9</td>
</tr>
<tr>
<td>1400</td>
<td>157.5</td>
<td>132.8</td>
<td>129.0</td>
<td>129.0</td>
</tr>
<tr>
<td>1800</td>
<td>138.8</td>
<td>113.9</td>
<td>110.1</td>
<td>110.1</td>
</tr>
<tr>
<td>2200</td>
<td>120.0</td>
<td>95.2</td>
<td>91.3</td>
<td>91.3</td>
</tr>
<tr>
<td>2600</td>
<td>101.2</td>
<td>76.6</td>
<td>72.7</td>
<td>72.7</td>
</tr>
<tr>
<td>3000</td>
<td>82.5</td>
<td>58.4</td>
<td>54.5</td>
<td>54.5</td>
</tr>
</tbody>
</table>

Linear Fit: \(y = -0.0471x + 223.5\)  
\(R^2 = 0.9900\)

### Table B.12: Average \(VF_{\theta^*}\) value (in units of average operation processing time) together with the associated linear fit for the four different staffing levels in the symmetric job shop.

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>60% Staffing</th>
<th>70% Staffing</th>
<th>80% Staffing</th>
<th>90% Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>223.2</td>
<td>181.7</td>
<td>154.8</td>
<td>142.9</td>
</tr>
<tr>
<td>600</td>
<td>204.0</td>
<td>162.6</td>
<td>136.2</td>
<td>124.5</td>
</tr>
<tr>
<td>1000</td>
<td>184.8</td>
<td>143.7</td>
<td>117.8</td>
<td>106.1</td>
</tr>
<tr>
<td>1400</td>
<td>165.7</td>
<td>124.9</td>
<td>99.3</td>
<td>87.4</td>
</tr>
<tr>
<td>1800</td>
<td>146.6</td>
<td>106.0</td>
<td>80.8</td>
<td>69.1</td>
</tr>
<tr>
<td>2200</td>
<td>127.5</td>
<td>87.5</td>
<td>62.4</td>
<td>51.1</td>
</tr>
<tr>
<td>2600</td>
<td>108.5</td>
<td>68.7</td>
<td>44.4</td>
<td>34.4</td>
</tr>
<tr>
<td>3000</td>
<td>89.6</td>
<td>50.3</td>
<td>27.9</td>
<td>22.3</td>
</tr>
</tbody>
</table>

Linear Fit: \(y = -0.0477x + 232.6\)  
\(R^2 = 0.9938\)
Appendix C. Sensitivity analysis

The assumption that the processing time of each operation is Uniformly distributed is a minimum information assumption, and was made because the processing times of operations in a job shop are usually highly dependent on the application at hand. To test the generality of the results obtained here, the same experiments were conducted using a Triangular distribution (with lower bound 1, upper bound 40, and mode 20.5) for the operation processing times. The Triangular distribution best characterizes the amount of information that can be expected from a job shop where more information is known about the operation processing times. For all problem classes almost identical results were obtained using Triangular distributions for all operation processing times compared with the results obtained using Uniform distributions for all operations processing times.

The same experiments were also conducted using the assumption that each job had between 9 and 16 operations, to test the sensitivity of the lower bound to the assumption about the number of operations per job. For the majority of problem classes, we again obtained almost identical results when compared with the results obtained using the assumption that each job had between 6 and 10 operations. Only in the symmetric job shop with 90% staffing and the asymmetric job shop with 70% staffing were the average differences $\text{VF}_{g^*} - \text{LB}_{g^*}$ and $\text{VF}_{\text{over}} - \text{LB}_{g^*}$ greater than in the corresponding problem classes when each job had between 6 and 10 operations.

References


