Allocating job-shop manpower to minimize $L_{\text{max}}$: Optimality criteria, search heuristics, and probabilistic quality metrics

Benjamin J. Lobo, Thom J. Hodgson, Russell E. King, Kristin A. Thoney, James R. Wilson

Abstract

We address questions raised by Lobo et al. in 2012 regarding the NP-hard problem of finding an optimal allocation of workers to machine groups in a job shop so as to minimize $L_{\text{max}}$, the maximum job latency. Lobo et al. formulated a lower bound on $L_{\text{max}}$ given a worker allocation, and an algorithm to find an allocation yielding the smallest such lower bound. In this article we establish optimality criteria to verify that a given allocation corresponds to a schedule that yields the minimum value of $L_{\text{max}}$. For situations in which the optimality criteria are not satisfied, we present the Heuristic Search Procedure (HSP), which sequentially invokes three distinct search heuristics, the Local Neighborhood Search Strategy (LNSS), Queuing Time Search Strategy 1 (QSS1), and Queuing Time Search Strategy 2 (QSS2), before delivering the best allocation encountered by LNSS, QSS1, and QSS2. HSP is designed to find allocations allowing a heuristic scheduler to generate schedules with a smaller value of $L_{\text{max}}$ than that achieved via the allocation yielding the final lower bound of Lobo et al. Comprehensive experimentation indicated that HSP delivered significant reductions in $L_{\text{max}}$. We also estimate a probability distribution for evaluating the quality (closeness to optimality) of an allocation delivered by a heuristic search procedure such as HSP. This distribution permits assessing the user’s confidence that a given allocation will enable the heuristic scheduler to generate its best possible schedule—i.e., the schedule with the heuristic scheduler’s smallest achievable $L_{\text{max}}$ value.

1. Introduction

Job shops in which both machines and workers form constraints are referred to in the literature as dual resource constrained (DRC) systems [1,2]. DRC systems more closely model real life by accounting for the constraints imposed on system operation by the limited availability of workers. In a job shop with $N$ jobs and $M$ machines on which to process the jobs, the $L_{\text{max}}$ single resource constrained job shop scheduling problem is to find a schedule that minimizes $L_{\text{max}}$, the maximum job lateness. This problem is known to be NP-hard [3]. Adding the worker constraint makes the problem even harder to solve, because it requires simultaneous optimization of the schedule and the allocation of workers to machine groups in the job shop.

Beginning with Nelson [4], DRC job shops have received considerable attention from the scheduling research community; see the reviews by Treleven [1], Gargeya and Deane [5], and Hottenstein and Bowman [6]. The previous research has focused on gauging the impact of the second constrained resource on job shop performance, and can be categorized for the most part by the worker assumptions. Examples of worker assumptions include worker allocation rules (where and when a worker can be transferred); worker skill set (e.g., the types of machines that a worker is capable of operating); worker efficiency; and in the more recent literature, worker learning, fatigue, and forgetfulness (i.e., skills and efficiencies that deteriorate if not used) [7–10]. However, little research could be found that determines an optimal way to allocate a fixed number of workers in a DRC job shop.

For the DRC job shop scheduling problem, Lobo et al. [11] recently formulated a lower bound on $L_{\text{max}}$ that accounts for both constrained resources. Given an allocation $\delta$ of workers to machine groups, the authors derived a lower bound $L_{\text{B}}$ on $L_{\text{max}}$ for that worker allocation. They also developed an algorithm to find a worker allocation $\delta^*$ that yields the smallest value of $L_{\text{B}}$ over all feasible values of $\delta$. The authors’ final lower bound on $L_{\text{max}}$, $L_{\text{B}}^*$, provides a benchmark against which heuristic solutions to this NP-Hard problem can be compared. However, the ultimate objective is to find an optimal allocation, defined to be an allocation associated with a schedule that minimizes $L_{\text{max}}$ for...
the job shop. Lobo et al. [11] observed randomly generated problem instances in which the schedule generated by the Virtual Factory (a heuristic scheduler developed by Hodgson et al. [12]) using an allocation other than $\mathcal{S}^*$ had a smaller $L_{\text{max}}$ value than that for the schedule based on allocation $\mathcal{S}^*$, showing that in general $\mathcal{S}^*$ is not guaranteed to be an optimal allocation of workers to machine groups.

In this article we present research that addresses the objective of finding an optimal or nearly optimal worker allocation for the DRC job shop scheduling problem through the following contributions:

1. Developing and justifying criteria which are easy to check in practice and which are sufficient to guarantee that the worker allocation $\mathcal{S}^*$ is optimal—i.e., $\mathcal{S}^*$ not only yields the smallest feasible value of $L_{\text{max}}$, but also corresponds to a schedule that minimizes $L_{\text{max}}$;
2. Formulating a heuristic search procedure to find optimal (or at least better) allocations in practical applications for which the optimality criteria of item 1 are not satisfied; and
3. Providing a method for probabilistically evaluating the quality (closeness to optimality) of a given allocation, and also assessing the user's degree of confidence that the Virtual Factory can use the given allocation to generate its best possible schedule.

If the optimality criteria of item 1 are not satisfied for a given problem instance, then we seek a "Virtual Factory–best" allocation using the search heuristic of item 2. Such an allocation enables the Virtual Factory to generate a schedule with the minimal value of $L_{\text{max}}$ that can be achieved by the Virtual Factory over all feasible allocations of workers to machine groups. In certain circumstances, the method of item 3 can be used to make the following statement: for a user-specified significance level $\alpha \leq 0.1$, the user's degree of confidence is approximately $100(1-\alpha)%$ that a given worker allocation will enable the Virtual Factory to generate a VF-best schedule for the DRC job shop scheduling problem at hand. It must be pointed out that the solution approach developed here would work with any other heuristic scheduler chosen, although the results obtained would differ based on the quality of the heuristic scheduler chosen.

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.

The rest of this article is organized as follows. Section 2 provides the background information required for precise specification of the DRC job shop scheduling problem. In Section 3 we formulate and justify criteria for checking whether a given worker allocation is optimal. In the case that neither optimality criterion is satisfied, in Section 4 we develop a number of heuristic strategies to search for an optimal allocation, and Section 5 provides a summary of the results of our experimental performance evaluation of the heuristics. In order to probabilistically evaluate the quality of the allocation delivered by the heuristics, in Section 6 we present a method for estimating certain probability distributions from which we can assess the user's degree of confidence that a given worker allocation will enable generation of a VF-best schedule. Section 7 recapitulates the main findings of this work and provides recommendations for future work. The Online Supplement to this article contains the following: (a) the proof of one of the optimality criteria found in Section 3; (b) a formulation of the confidence-interval estimation method used to evaluate the performance of the proposed heuristic search procedure provided in Section 5; and (c) the full set of results supporting the experimental performance evaluation summarized in Sections 5 and 6.

2. Background on the DRC job shop scheduling problem

In this article we make the following assumptions about the workers in a DRC job shop:

- 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 a time; moreover, the worker must be present at a machine whilst processing of a job is underway.
- The allocation of workers to machine groups, once determined, is fixed.
- Every worker can be assigned to any machine group in the job shop.

Machine groups arise naturally in a job shop; the machines that perform similar operations will typically be located in physical proximity of each other and will require the same basic skills and expertise to operate.

The problem addressed in this article is denoted as the $J_{\text{M}}/W/L_{\text{max}}$ DRC job shop scheduling problem [11], and is defined as follows. The DRC job shop has a total of $N$ jobs and $M$ machines on which to process those jobs. Every job $j$ has a routing 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 $m_g$ denotes the $i$th machine group consisting of the set of machines $\mathcal{M}_i$ for $i = 1, \ldots, S$, with $S$ denoting the number of machine groups in the job shop. Of the $W$ workers in the job shop, $w_i$ workers are assigned to $m_g$, so that $1 \leq w_i \leq |\mathcal{M}_i|$ (where $|\mathcal{M}_i|$ denotes the size of the set $\mathcal{M}_i$ for $i = 1, \ldots, S$; and we have $\sum_{i=1}^{S} w_i = W < M = \sum_{i=1}^{S} |\mathcal{M}_i|$). The objective is to determine an allocation of workers to machine groups in the job shop that corresponds to a schedule yielding the minimal value of $L_{\text{max}}$. Because the $J_{\text{M}}/W/L_{\text{max}}$ DRC job shop scheduling problem is NP-Hard, a heuristic scheduler is used to generate the schedule given a particular allocation $\mathcal{S}$ of workers to machine groups.

The Virtual Factory was developed by Hodgson et al. [12] as part of their solution to the $J_{\text{M}}/L_{\text{max}}$ single resource constrained job shop scheduling problem. Incorporating a transient, deterministic simulation of a job shop, the Virtual Factory is based on an approach proposed by Vepsäläinen and Morton [13]. The Virtual Factory is an iterative procedure that sequences jobs using a slack calculation on the first iteration. On each subsequent iteration, the Virtual Factory sequences the jobs using an iteratively revised slack calculation that includes queuing time estimates from the previous iteration. Iterations continue until a previously computed lower bound on $L_{\text{max}}$ is achieved or until the number of iterations reaches a prespecified limit; then the Virtual Factory delivers the best schedule encountered over all iterations. We chose the Virtual Factory as the heuristic scheduler because of its proven track record in successfully generating optimal or near-optimal schedules to job shop scheduling problems for which the primary objective is to minimize $L_{\text{max}}$ [14–19], but the approach presented in this article would work with other heuristic schedulers.
Lobo et al. [11] determine a lower bound on $L_{\text{max}}$ for a given allocation $\mathcal{A}$ of workers to machine groups as follows. First, a network flow approach is used to find a lower bound on $L_{\text{max}}$ for each individual machine group; then the lower bound $L_{B_j}$ on $L_{\text{max}}$ for the entire DRC job shop is taken to be the maximum of the respective lower bounds on $L_{\text{max}}$ for the individual machine groups. Next the authors' Lower Bound Search Algorithm (LBSA) is used to find the smallest such lower bound on $L_{\text{max}}$ for the job shop taken over all feasible allocations of workers to machine groups, i.e., $L_{B_j}$. LBSA delivers the worker allocation $\mathcal{A}^*$ together with the index $k$ of a constraining machine group. Given a particular allocation $\mathcal{A}$ of workers to machine groups, a constraining machine group is a machine group whose individual lower bound on $L_{\text{max}}$ is equal to the lower bound $L_{B_j}$ on $L_{\text{max}}$ for the entire job shop.

**Remark 1.** For each constraining machine group associated with a given allocation $\mathcal{A}$, that machine group's lower bound on $L_{\text{max}}$ coincides with $L_{B_j}$, the lower bound on $L_{\text{max}}$ for the entire DRC job shop based on allocation $\mathcal{A}$ as detailed in the preceding paragraph. Therefore a constraining machine group is a good candidate for having an additional worker reassigned to it from some other nonconstraining machine group, so that the resulting new allocation coincides with $L_{B_j}$.

Section 3 contains a more detailed explanation of the termination conditions for LBSA. Lobo et al. [11] prove that LBSA is guaranteed to find an allocation yielding the smallest $L_{B_j}$ value, where several such allocations may exist; and in some (rare) cases, there may be several constraining machine groups associated with the allocation delivered by LBSA.

3. Optimality criteria

For each feasible allocation $\mathcal{A}$, let $L_{B_j}$ denote the associated lower bound on $L_{\text{max}}$ for the job shop as defined in Lobo et al. [11] and in Section 2 of this article. Similarly, we let $V_{F_\mathcal{A}}$ denote the value of $L_{\text{max}}$ resulting from the schedule generated by the Virtual Factory using allocation $\mathcal{A}$. A VF-best allocation (denoted by $\mathcal{A}^{\text{VF-B}}$) is defined to be one for which a schedule generated by the Virtual Factory will have the minimum value of $L_{\text{max}}$ that can be achieved using the Virtual Factory with all feasible allocations of workers to machine groups

$$\mathcal{A}^{\text{VF-B}} = \arg \min \left\{ V_{F_\mathcal{A}} : \mathcal{A} \text{ is a feasible allocation, i.e., } 1 \leq w_i(\mathcal{A}) \leq |M_k| \right\}$$

where $w_i(\mathcal{A})$ denotes the number of workers allocated to machine group $i$ under allocation $\mathcal{A}$ for $i = 1, \ldots, S$. We let $V_{F_\mathcal{A}}$ denote the corresponding VF-best value of $L_{\text{max}}$.

**Remark 2.** If the right-hand side of Eq. (1) defines a set with two or more elements, then allocation $\mathcal{A}^{\text{VF-B}}$ is understood to be an arbitrary member of that set. Note that a VF-best allocation is not necessarily an optimal allocation, and that an optimal allocation is not necessarily a VF-best allocation.

**Theorem 1.** If the allocation $\mathcal{A}$ satisfies

$$V_{F_\mathcal{A}} = L_{B_j}$$

then $\mathcal{A}$ is an optimal allocation—that is, associated with allocation $\mathcal{A}$ there is a schedule that yields the minimum possible value of $L_{\text{max}}$ for the job shop; moreover, $\mathcal{A}$ is a VF-best allocation.

**Proof.** Eq. (2) immediately implies that $\mathcal{A}$ is any feasible allocation, then Eq. (2) and Proposition 2 of Lobo et al. [11] ensure that $V_{F_\mathcal{A}} = L_{B_j}$, showing that $\mathcal{A}$ is also a VF-best allocation.

Next we formulate a sufficient condition for the allocation $\mathcal{A}^*$ delivered by LBSA to be both an optimal and a VF-best allocation. For $i = 1, \ldots, S$, let $L_{B_{mg}}(w_i(\mathcal{A}^*))$ denote the lower bound on $L_{\text{max}}$ for machine group $i$ when $w_i(\mathcal{A}^*)$ workers are assigned to that machine group, and let

$$K(\mathcal{A}^*) = \{i : 1 \leq i \leq S \text{ and } L_{B_{mg}}(w_i(\mathcal{A}^*)) = L_{B_j}\}.$$  

denote the index-set of constraining machine groups for allocation $\mathcal{A}^*$. If LBSA terminates with allocation $\mathcal{A}^*$ and with the currently selected constraining machine group $k \in K(\mathcal{A}^*)$, then we define the neighborhood of $\mathcal{A}^*$ with respect to the constraining machine group $k$ as follows:

$$N(\mathcal{A}^*, k) = \left\{ \phi : w_j(\phi) = w_j(\mathcal{A}^*) + 1 \text{ and } w_k(\phi) = w_k(\mathcal{A}^*) - 1 \text{ for some } j \neq k; \text{ and } w_j(\phi) + w_k(\phi) = w_j(\mathcal{A}^*) + w_k(\mathcal{A}^*) \right\}$$

Moreover, we define the associated neighborhood-optimal allocation

$$\phi^*(\mathcal{A}^*, k) = \arg \min L_{B_{\phi^*}} : \phi \in N(\mathcal{A}^*, k)$$

with the corresponding lower bound on $L_{\text{max}}$

$L_{B_{\phi^*(\mathcal{A}^*, k)}} \equiv \min L_{B_{\phi}} : \phi \in N(\mathcal{A}^*, k)$.

**Remark 3.** Throughout this article, $\mathcal{A}$ is used to denote an arbitrary allocation currently under discussion, while the symbol $\mathcal{A}^*$ is reserved to denote the allocation delivered by LBSA whose $L_{\text{max}}$ value $L_{B_j}$ is the minimum value of $L_{B_j}$ taken over all feasible allocations $\mathcal{A}$. Similarly, other lower case Greek letters are used strategically to distinguish between different types of allocations. For example, $\phi$ is used throughout this article to denote an arbitrary allocation in the neighborhood $N(\mathcal{A}^*, k)$ of $\mathcal{A}^*$ defined by constraining machine group $k$, while $\phi^*(\mathcal{A}^*, k)$ is used to denote an allocation in that neighborhood whose $L_{\text{max}}$ value $L_{B_{\phi^*(\mathcal{A}^*, k)}}$ is the minimum value of $L_{B_{\phi}}$ taken over all feasible allocations $\phi$ in that neighborhood. In general, a superscript is appended to a lower case Greek letter to denote an allocation with some relevant property; for example, $\mathcal{A}^{\text{VF-B}}$ is used to denote...
an allocation delivered by the Heuristic Search Procedure (HSP) detailed in Section 4.

Remark 4. On the right-hand side of Eq. (5), we take \( \text{LB}_\phi \equiv \infty \) if allocation \( \phi \) is infeasible, where the feasibility condition is given on the right-hand side of Eq. (1). Moreover if the right-hand side of Eq. (5) specifies a set consisting of more than one neighborhood-optimal allocation, then \( \phi^*(\mathcal{G}, k) \) is understood to be an arbitrary member of that set.

Theorem 2. If LBSA terminates with an allocation \( \mathcal{G}^* \) by satisfying Condition 2 with a single constraining machine group \( k \) (i.e., \( K(\mathcal{G}^*) = \{k\} \) such that

\[
\text{LB}_\phi < \text{LB}_{\phi^*(\mathcal{G}, k)} \quad \text{and} \quad \text{VF}_{\phi} < \text{VF}_{\phi^*(\mathcal{G}, k)}.
\]

then \( \mathcal{G}^* \) is an optimal allocation; moreover, \( \mathcal{G}^* \) is a VF-best allocation.

The proof of Theorem 2 appears in Section A1 of the Online Supplement.

Remark 5. In Theorem 2 the assumption of a single constraining machine group for allocation \( \mathcal{G}^* \) might appear to be overly restrictive, but in our experience this is not the case. Of the 16,000 randomly generated instances of a symmetric job shop used in our experimentation, there were 11,712 cases in which LBSA delivered allocation \( \mathcal{G}^* \) by terminating because Condition 2 was satisfied; and of these latter cases, there were only 128 cases (i.e., 1.05% of the relevant problem instances) in which LBSA terminated with more than one constraining machine group. Of the 16,000 randomly generated instances of an asymmetric job shop used in our experimentation, there were 4974 cases in which LBSA delivered allocation \( \mathcal{G}^* \) by terminating because Condition 2 was satisfied; and of these latter cases, there were only 34 cases (i.e., 0.68% of the relevant problem instances) in which LBSA terminated with more than one constraining machine group. Thus, we believe that Theorem 2 can be useful in practice for rapid verification that \( \mathcal{G}^* \) is optimal.

Table 1 shows, for both an asymmetric and symmetric job shop, and for staffing levels of 60%, 70%, 80%, and 90%, the percentage of randomly generated problem instances in which allocation \( \mathcal{G}^* \) satisfied the hypotheses of either Theorem 1 or Theorem 2 (out of 200 randomly generated problem instances for each combination of type of job shop, percent staffing, and due-date range). An in-depth description of the experimental job shop can be found in Lobo et al. [11] and in Section 5.1 of this article. This table indicates that symmetric job shop problems with 70%, 80%, and 90% staffing are especially difficult types of problems to solve, as are asymmetric job shop problems with 70% staffing.

We observed that there are problem instances with the following anomalous properties: (a) allocation \( \mathcal{G}^* \) does not satisfy either of the optimality criteria (i.e., the hypotheses of either Theorem 1 or Theorem 2); and (b) there is no feasible allocation \( \beta \) that satisfies the hypothesis of Theorem 1. This observation motivated the heuristic search strategies presented in the following section, which search for a VF-best allocation when allocation \( \mathcal{G}^* \) delivered by LBSA does not satisfy either optimality criteria presented in this section.

4. Heuristic search strategies

If allocation \( \mathcal{G}^* \) does not satisfy either of the optimality criteria, then other promising allocations must be identified and evaluated. If we are lucky, then we will encounter a feasible allocation \( \beta \neq \mathcal{G}^* \) such that \( \text{VF}_\beta = \text{LB}_\beta \). Theorem 1 is satisfied. More often, however, we must hope that our search returns a VF-best allocation. In Section 4.1 we present a partial enumeration strategy for finding a VF-best allocation, and we summarize some computational results illustrating the impracticality of this approach and underscoring the need for effective heuristics to search for a VF-best allocation. Section 4.2 details a heuristic that searches the local neighborhood of allocation \( \mathcal{G}^* \), and Section 4.3 presents two heuristics that rely on queuing time estimates to guide the search for a VF-best allocation. Section 4.4 describes the combined heuristic search strategy HSP that incorporates the three heuristics of Sections 4.2 and 4.3 and that is used in all experimentation reported in Section 5.

4.1. A partial enumeration strategy for finding a VF-best allocation

For a given problem, to guarantee that an allocation is a VF-best allocation, we must use the Virtual Factory to evaluate each feasible allocation \( \beta \) for which \( \text{LB}_\beta < \text{VF}_\beta \). The partial enumeration strategy outlined below attempts to minimize the number of allocations that must be evaluated to find a VF-best allocation (which is sometimes also an optimal allocation). This strategy progressively reduces the set of feasible worker allocations that remain to be evaluated by iteratively reducing the upper bound \( \text{VF}_\beta \) on acceptable values of \( \text{LB}_\beta \) whenever a "VF-better" allocation \( \beta \) is encountered—i.e., whenever a newly encountered allocation \( \beta \) yields the smallest value \( \text{VF}_\beta \) encountered so far.

For each machine group \( j (1 \leq i \leq S) \) in the job shop, we first compute and store \( \text{LB}_{\text{mg}}(w_i) \) for \( 1 \leq w_i \leq |M_j| \). This information can be used to determine the full set of feasible allocations that yield the smallest \( \text{LB}_{\beta_j} \) value for the DRC job shop.

\[
\Psi = \arg \min \{ \text{LB}_\beta : \beta \text{ feasible} \} = \{ \beta : \text{LB}_\beta = \text{LB}_{\beta_j} \}.
\]

The Virtual Factory can then be used to find an allocation

\[
\psi^* = \arg \min \{ \text{VF}_\beta : \psi \in \Psi \}, \quad \text{so that } \text{VF}_{\psi^*} = \min \{ \text{VF}_\beta : \psi \in \Psi \}.
\]

Each allocation \( \gamma \in \Gamma \) must be evaluated using the Virtual Factory to check for the condition \( \text{VF}_\gamma < \text{VF}_{\psi^*} \), showing that \( \gamma \) is a better allocation than \( \psi^* \). We can establish an iteratively reduced upper bound on \( \text{LB}_\beta \) that requires only a subset of \( \Gamma \) to be evaluated as follows: each time an allocation \( \beta \in \Gamma \) is encountered such that \( \text{VF}_\beta \) is strictly less than the smallest \( \text{VF} \) value found so far, from that point on we evaluate only allocations \( \gamma \in \Gamma \) for which \( \text{LB}_\gamma < \text{LB}_\beta < \text{VF}_\gamma \). A VF-best allocation is then given by Eq. (1) so that

\[
\Psi_{\text{VF}} = \min \{ \text{VF}_\beta : \beta \text{ feasible} \} = \min \{ \text{VF}_\beta : \beta \in \Gamma \cup \{ \psi^* \} \}.
\]

In general, the progressively reduced upper bound used in our partial enumeration strategy ensures that we stop substantially short of a complete enumeration of all feasible worker allocations.

<table>
<thead>
<tr>
<th>Due-date range</th>
<th>Asymmetric job shop</th>
<th>Symmetric job shop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staffing level</td>
<td>60% 70% 80% 90%</td>
<td>60% 70% 80% 90%</td>
</tr>
<tr>
<td>200</td>
<td>67.0 56.0 99.5 99.5</td>
<td>67.0 48.0 18.5 66.0</td>
</tr>
<tr>
<td>600</td>
<td>69.5 45.5 97.5 97.5</td>
<td>72.5 47.5 17.0 56.5</td>
</tr>
<tr>
<td>1000</td>
<td>67.0 40.5 92.5 92.0</td>
<td>75.0 47.5 13.0 46.0</td>
</tr>
<tr>
<td>1400</td>
<td>65.0 35.0 87.5 87.0</td>
<td>74.5 41.5 9.0 42.5</td>
</tr>
<tr>
<td>1800</td>
<td>65.0 35.5 86.5 86.5</td>
<td>75.0 38.5 6.5 36.0</td>
</tr>
<tr>
<td>2200</td>
<td>57.0 32.5 81.0 82.0</td>
<td>76.5 32.5 6.0 23.0</td>
</tr>
<tr>
<td>2600</td>
<td>55.5 28.5 75.5 76.0</td>
<td>73.0 27.0 2.5 14.5</td>
</tr>
<tr>
<td>3000</td>
<td>50.5 27.0 70.0 70.0</td>
<td>67.5 20.0 0.5 4.5</td>
</tr>
</tbody>
</table>

Table 1
Percentage of randomly generated problem instances for which allocation \( \mathcal{G}^* \) satisfies the optimality criteria of Theorem 1 or Theorem 2.
Table 2
Sample mean \( T \) and standard deviation \( S_T \) of the execution time \( T \) taken (in seconds) for the partial enumeration strategy 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></td>
<td>( T )</td>
<td>( S_T )</td>
<td>( T )</td>
<td>( S_T )</td>
</tr>
<tr>
<td>200</td>
<td>273.6</td>
<td>1029.4</td>
<td>288.2</td>
<td>639.7</td>
</tr>
<tr>
<td>600</td>
<td>95.5</td>
<td>122.3</td>
<td>191.1</td>
<td>328.9</td>
</tr>
<tr>
<td>1000</td>
<td>83.8</td>
<td>75.8</td>
<td>195.0</td>
<td>309.6</td>
</tr>
<tr>
<td>1400</td>
<td>96.3</td>
<td>103.4</td>
<td>182.0</td>
<td>299.2</td>
</tr>
<tr>
<td>1800</td>
<td>78.5</td>
<td>64.2</td>
<td>211.2</td>
<td>365.5</td>
</tr>
<tr>
<td>2200</td>
<td>75.1</td>
<td>58.1</td>
<td>261.7</td>
<td>426.8</td>
</tr>
<tr>
<td>2600</td>
<td>72.0</td>
<td>51.2</td>
<td>365.8</td>
<td>662.7</td>
</tr>
<tr>
<td>3000</td>
<td>79.9</td>
<td>112.8</td>
<td>519.7</td>
<td>1043.6</td>
</tr>
</tbody>
</table>

For a randomly generated instance of a symmetric job shop scheduling problem, let the random variable \( T \) denote the execution time needed to enumerate the allocation search space for that problem. The sample statistics in Table 2 were obtained using Table 1.

4.2. Local neighborhood search strategies (LNSS)

Recall that the Lower Bound Search Algorithm (LBSA) terminates if either of two termination conditions is satisfied as discussed at the end of Section 2 and in a remark in the Online Supplement. Two different search strategies are developed in Sections 4.2.1 and 4.2.2, based on which condition is satisfied when LBSA terminates. Both search strategies evaluate allocations in what can be termed the local neighborhood of allocation \( \theta^* \), and together they constitute what is referred to as the Local Neighborhood Search Strategy (LNSS).

4.2.1. Type-\( \alpha \) allocations

If the LBSA terminates because Condition 1 is satisfied, then no additional workers can be assigned to machine group \( k \), the currently selected constraining machine group of allocation \( \theta^* \). Since nothing further can be done to decrease the lower bound on \( L_{\max} \) for machine group \( k \), we consider additional worker allocations \( \theta^j \) derived from \( \theta^* \) that satisfy the condition \( LB_{\theta^j} \leq LB_{\theta^*} \), and that are called type-\( \alpha \) allocations. Algorithm 1 is designed to find a type-\( \alpha \) allocation \( \theta^j \) that has one of the following properties:

1. \( \theta^j \) is both an optimal and a VF-best allocation; or
2. \( VF_{\theta^j} \) is the smallest value of \( L_{\max} \) delivered by the Virtual Factory taken over all type-\( \alpha \) worker allocations encountered during the execution of Algorithm 1.

A type-\( \alpha \) allocation is found by allocating workers to minimize the lower bound on \( L_{\max} \) of the machine group \( i \) with the largest value of \( LB_{\theta^j} \), ignoring any machine groups that satisfy Condition 1 of LBSA since the respective lower bounds on \( L_{\max} \) for these machine groups cannot be improved. Any machine group \( i \) that has a full complement of \( |M_i| \) workers assigned to it may not donate a worker, based on the assumption that unassigning a worker from such a machine group would cause it to become constraining. In addition, once a machine group \( j \) donates a worker to another machine group, machine group \( j \) may not receive an additional worker. Similarly, once a machine group \( j \) receives a worker from another machine group, machine group \( j \) may not donate a worker.

Algorithm 1 terminates when one of the following conditions occurs: (i) the maximum search time has elapsed; (ii) a type-\( \alpha \) allocation \( \theta^j \) is encountered that satisfies \( VF_{\theta^j} = LB_{\theta^j} \); so that \( \theta^j \) is both an optimal and a VF-best allocation; or (iii) the type-\( \alpha \) allocation \( \theta^j \) with the smallest possible value of \( VF_{\theta^j} \) has been found.

Algorithm 1. Algorithm to find Type-\( \alpha \) allocations.

\[ \theta^j \leftarrow \theta^* \]

\[ VF_{\theta^j} \leftarrow VF_{\theta^*} \]

For each \( j \in \{1, \ldots, S\} \)

- If \( \theta_j(\theta^j) = |M_j| \)
  - \( \text{MayDonate}[j] \leftarrow \text{false} \)
  - \( \text{MayReceive}[j] \leftarrow \text{false} \)
- Else if \( \theta_j(\theta^j) = 1 \)
  - \( \text{MayDonate}[j] \leftarrow \text{false} \)
  - \( \text{MayReceive}[j] \leftarrow \text{true} \)
- Else
  - \( \text{MayDonate}[j] \leftarrow \text{true} \)
  - \( \text{MayReceive}[j] \leftarrow \text{false} \)

\[ \lambda \leftarrow 0 \]

\[ \theta^j \leftarrow \theta^* \]

While \( (\text{SearchTimeElapsed} < 45 \text{ sec}) \)

For each \( j \in \{1, \ldots, S\} \)

- If \( \text{MayDonate}[j] = \text{true} \) and \( LB_{\theta^j}(\theta^j) - 1 \leq LB_{\theta^*} \)
  - If there exists \( mg, \text{ such that } mg \neq mg_j \) and \( \text{MayReceive}[j] = \text{true} \)
    - \( g^{j+1} \leftarrow g^j \)
    - \( w_j(g^{j+1}) = w_j(\theta^j) + 1 \)
    - \( \theta_j(g^{j+1}) = \theta_j(\theta^j) - 1 \)
    - \( \lambda \leftarrow \lambda + 1 \)
    - \( \text{MayDonate}[j] \leftarrow \text{false} \)
    - \( \text{MayReceive}[j] \leftarrow \text{false} \)
  - Evaluate \( \theta^j \) using the Virtual Factory.
  - If \( VF_{\theta^j} < VF_{\theta^*} \)
    - \( \theta^j \leftarrow \theta^* \)
    - \( VF_{\theta^*} \leftarrow VF_{\theta^*} \)
  - If \( VF_{\theta^j} = LB_{\theta^*} \)
    - Deliver the type-\( \alpha \) allocation \( \theta^j \) with VF-best value \( VF_{\theta^*} \) and stop.

Deliver the type-\( \alpha \) allocation \( \theta^j \) with the smallest possible value \( VF_{\theta^*} \) and stop.

4.2.2. Type-\( \beta \) allocations

If LBSA terminates at allocation \( \theta^* \) with constraining machine group \( k \in K(\theta^j) \) because Condition 2 is satisfied, then the neighborhood \( N(\theta^*, k) \) of \( \theta^* \) defined by Eq. (4) consists of up to 51 allocations, where for \( j \in \{1, \ldots, S\} \), the \( j \)-th neighbor \( \phi_j \) of \( \theta^* \) is defined in terms of \( \theta^j \) by the worker reassignment \( w_j(\phi_j) = w_j(\theta^j) - 1 \) and \( w_k(\phi_j) = w_k(\theta^j) + 1 \), with \( w_i(\phi_j) = w_i(\theta^j) \).
for \(i \neq j, k\) (assuming that it is feasible to obtain \(\phi^i\) from \(\beta^*\) by reassigning one worker from \(mg_i\) to \(mg_{\beta^*}\); otherwise \(\phi^i\) does not exist). For each allocation \(\phi^i \in \mathcal{N}(\beta^*, k)\), we define that allocation's reduced neighborhood

\[
\mathcal{N}(\phi^i, \beta^*, k) = \left\{ \phi^m \in \mathcal{N}(\beta^*, k) \mid w_{B}(\phi^m) = w_{B}(\phi^i) \right\},
\]

where \(w_{B}(\phi^i) = w_{B}(\phi^j) - 1\) and \(w_{B}(\phi^m) = w_{B}(\phi^i) + 1\) for some \(u \in \{j, k\}\) and \(w_{B}(\phi^i) = w_{B}(\phi^j)\) for \(i \neq j, k\).

\[
(7)
\]

In other words if allocation \(\phi^i\) exists, then for \(u \in \{1, \ldots, S\}\), the \(u\)th neighbor of allocation \(\phi^i\) is defined by the worker reassignment \(w_{B}(\phi^i) = w_{B}(\phi^j) - 1\) and \(w_{B}(\phi^m) = w_{B}(\phi^i) + 1\), with \(w_{B}(\phi^m) = w_{B}(\phi^j)\) for \(i \neq j, k\) (assuming that it is feasible to obtain \(\phi^m\) from \(\phi^i\) by reassigning one worker from \(mg_i\) to \(mg_{\beta^*}\); otherwise, \(\phi^m\) does not exist).

The worker allocations defined by the right-hand side of Eq. (7) are called type-\(\beta\) allocations; and associated with \(\beta^*\) and the currently constraining machine group \(k\), we have the full set of type-\(\beta\) allocations

\[
B(\beta^*, k) = \bigcup_{\phi^i \in \mathcal{N}(\beta^*, k)} \mathcal{N}(\phi^i, \beta^*, k).
\]

For each type-\(\beta\) allocation \(\phi^m\) that belongs to the reduced neighborhood \(\mathcal{N}(\phi^i, \beta^*, k)\) of the allocation \(\phi^i \in \mathcal{N}(\beta^*, k)\), we have

\[
\text{LB}_{\phi^m} = \text{LB}_{\phi^i} = \text{LB}_{\phi}(w_{B}(\phi^i)) = \text{LB}_{\phi}(w_{B}(\phi^j) - 1);
\]

and thus allocation \(\phi^i\) and all the type-\(\beta\) allocations in its reduced neighborhood \(\mathcal{N}(\phi^i, \beta^*, k)\) are equally promising candidates to be evaluated by the Virtual Factory. Our search strategy is to evaluate each neighbor of allocation \(\beta^*\) and the type-\(\beta\) allocations associated with that neighbor using the Virtual Factory, where the evaluations are performed in order of increasing value of the lower bound on \(L_{\text{max}}\) for the candidate allocations. The evaluations continue until we find an optimal allocation, or the maximum search time has elapsed; and finally we deliver the allocation \(\beta^*\) whose associated value \(\text{VF}_{\beta^*}\) is the smallest VF-value taken over all the allocations in \(\mathcal{N}(\beta^*, k) \cup B(\beta^*, k)\) that were evaluated by the Virtual Factory.

**Remark 6.** Depending on whether LBSA terminates by satisfying either Condition 1 or Condition 2, LNSS respectively delivers either \(\beta^*\) or \(\beta^\dagger\), the best allocation of the relevant type that was encountered in the execution of LNSS.

4.3. Queuing time search strategies

In the course of searching for the best schedule given an allocation \(\beta\), the Virtual Factory computes the estimated queuing time of each job at each machine visited by that job. For the job with maximal latency \(L_{\text{max}}\) (henceforth simply called the \(L_{\text{max}}\) job), the Virtual Factory computes the job's estimated queuing time in a machine group by summing the estimated queuing times for that job at each machine visited by the job on its route through the machine group. We use these estimated machine-group queuing times to identify the machine groups for which the assignment of an additional worker may reduce \(L_{\text{max}}\), as well as the machine groups for which the unassignment of a worker will probably not increase \(L_{\text{max}}\). Note that both queuing time search strategies presented here require an initial “seed” allocation whose estimated queuing times can be used to determine the next allocation searched.

4.3.1. Queuing time search strategy 1 (QSS1)

In the first queuing time search strategy (referred to as QSS1), after an allocation has been evaluated, the estimated machine-group queuing times for the \(L_{\text{max}}\) job are used to determine a new allocation as follows. Since each machine group must have at least one worker assigned to it, 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 \(\text{Proc}_N\) denote the sum of the processing times of all the jobs that must be processed in the job shop. Let \(L_{\text{max}}\) denote the estimated queuing time in machine group \(i\) for the \(L_{\text{max}}\) job, and let \(L_{\text{max}}\) denote the estimated total queuing time in the job shop for the \(L_{\text{max}}\) job. Then the additional number of workers that will be allocated to machine group \(i\) is

\[
\text{min}\left\{\left[\frac{\text{Proc}_i + L_{\text{max}}}{\text{Proc}_N + L_{\text{max}}}ight] \cdot (W - S) \mid |M_i| - 1\right\}.
\]

(8)

Because of the rounding down in Eq. (8) and the constraint on the total number of workers to be assigned, there may be workers that still need to be assigned after the \(S\) machine groups have been allocated workers using Eq. (8). In this case, the remaining workers are assigned sequentially to the machine groups with the largest values of the remainder

\[
\left\{\frac{\text{Proc}_i + L_{\text{max}}}{\text{Proc}_N + L_{\text{max}}} \cdot (W - S) - \left[\frac{\text{Proc}_i + L_{\text{max}}}{\text{Proc}_N + L_{\text{max}}} \cdot (W - S)\right]\right\}.
\]

(9)

that do not already have \(|M_i|\) workers assigned to them, where \(i = 1, \ldots, S\). If the resulting allocation \(\beta^\ddagger\) has not already been evaluated, then it is evaluated using the Virtual Factory and new estimated queuing times are obtained. The new estimated queuing times are then used in Eqs. (8) and (9) to obtain a new candidate allocation; and this process is repeated until one of the following termination conditions is satisfied: (i) the current allocation \(\beta^\ddagger\) has already been evaluated; (ii) the current allocation \(\beta^\ddagger\) satisfies the optimality criterion \(\text{VF}_{\beta^\ddagger} = \text{LB}_{\beta^\ddagger}\); or (iii) the maximum search time has elapsed. When QSS1 terminates, it delivers the allocation \(\beta^\ddagger\) whose associated value \(\text{VF}_{\beta^\ddagger}\) is the smallest such VF-value taken over all allocations evaluated by the Virtual Factory during the course of executing QSS1.

4.3.2. Queuing time search strategy 2 (QSS2)

The second queuing time search strategy (referred to as QSS2) uses the estimated queuing time information to modify the base allocation (defined as the allocation whose associated VF-value is the smallest encountered so far) by selecting a machine group to have one worker unassigned from it, and selecting a different machine group to have a worker reassigned to it. The receiving machine group \(j\) is the machine group that lies on the \(L_{\text{max}}\) job’s route, has the largest \(L_{\text{max}}\) value, and can receive a worker (i.e., the machine group \(i\) currently has fewer workers assigned to it than machines in the group). The donating machine group \(j\) is the machine group that does not lie on the \(L_{\text{max}}\) job’s route, has the smallest \(\text{Proc}_j\), and can donate a worker (i.e., it currently has more than one worker assigned to it). The motivation behind this approach to modifying the base allocation is that the receiving machine group appears to have the greatest need for one additional worker, while the donating machine group appears to be most likely to perform at an acceptable level after the loss of one worker.

For the modified allocation \(\beta^\ddagger\), if the associated value \(\text{VF}_{\beta^\ddagger}\) is less than the smallest VF value found so far, then \(\beta^\ddagger\) becomes the base allocation. On the other hand if \(\text{VF}_{\beta^\ddagger}\) exceeds the smallest VF value found so far, then \(\beta^\ddagger\) is discarded, the base allocation remains unchanged, and a new allocation is derived from the base allocation by selecting a new donating machine group and a new receiving machine group. For a given base allocation, each machine group can be either a donating or receiving machine group at most once. Iterations of QSS2 continue until one of the following termination conditions is satisfied: (i) the current modified allocation \(\beta^\ddagger\) has
already been evaluated; (ii) the current modified allocation \( \vartheta^2 \) satisfies the optimality criterion \( V^{\text{F,LB}} = L^{\text{P-L}} \); or (iii) the maximum search time has elapsed. When QSS2 terminates, it delivers the allocation \( \vartheta^2 \) whose associated value \( V^{\text{F,HB}} \) is the smallest such VF-value taken over all allocations evaluated by the Virtual Factory during the course of executing QSS2.

4.4. Combined heuristic search procedure

The three heuristic search strategies are combined into a single heuristic search procedure (HSP) as follows. Procedure HSP starts with LNNS, which is initialized using the “seed” allocation \( \vartheta_s \). When LNNS terminates, QSS1 is initialized using as the “seed” allocation either \( \vartheta^2 \) or \( \vartheta^0 \) as returned by LNNS. When QSS1 terminates, QSS2 is initialized using the “seed” allocation \( \vartheta^1 \) returned by QSS1. When QSS2 terminates, it returns allocation \( \vartheta^2 \). After all stages of the combined heuristic search procedure are complete, HSP delivers \( \vartheta^{\text{HSP}} \), the allocation yielding the smallest VF-value taken over all allocations encountered during the course of executing HSP. Alternative heuristic search procedures were tried (e.g., using allocation \( \vartheta^2 \) as the “seed” allocation for the queuing time search strategies and changing the order of executing the three search strategies); however the version of HSP presented here was found to provide the best results overall.

5. Experimentation

5.1. Experimental design

Experiments were performed to explore how HSP performed in different types of DRC job shops, with different staffing levels, and over a variety of due-date ranges. The experimental job shop used here is the same as that used by Lobo et al. [11]. The job shop has 80 machines split into 10 machine groups, where each machine group has eight machines. There are 1200 jobs that require processing. The due-date of each job is a function of the due-date range, which varies from 200 to 3000 in increments of 400. (Preliminary experiments showed that, when the maximum due-date exceeded 3000, the maximum flow time in the job shop was less than the greatest due-date, an unrealistic scenario.) The processing time of each operation is a random variable sampled from the discrete uniform distribution on the integers \([1, 2, ..., 40]\); Lobo et al. [11] found that job shop performance was insensitive to this assumption. Every job has between 6 and 10 operations, and at most 3 of those operations can occur in the same machine group (not necessarily consecutively).

Demirkol et al. [20] use the parameters of due-date range and the expected number of tardy jobs to determine job due-dates for benchmark job shop problems in which \( l_{\text{max}} \) is the objective to be minimized. However, Hodgson et al. [14] find that if the due-date “range is held constant, the tightness of the due-dates does not affect the optimal sequence.” Thus in this article the difficulty of problems being solved is varied by varying the due-date range. Four different overall job shop staffing levels are considered, namely, 60%, 70%, 80%, and 90% (a 90% staffing level corresponds to there being 72 workers available for allocation to the 80 machines).

The type of job shop, reflected in the symmetric (balanced) and asymmetric (unbalanced) loading of the machine groups, also forms part of the experimental design. In the symmetric version, every machine group has an equal probability of being on a job’s route, which results in each of the machine groups seeing 10% of the total workload on average. In the asymmetric version, some machine groups have a greater probability of being on a job’s route than others, so that four machine groups each see more than 10% of the total workload on average, while the other six machine groups each see less than 10% of the total workload on average (see Lobo et al. [11] for the exact percentages).

The combination of the job shop type, the 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; and for \( i = 1, ..., 200 \), the \( i \)th randomly generated problem instance is the same problem instance that is used to compare the performance of the allocations \( \vartheta^{\text{HSP}} \) and \( \vartheta^* \) at each staffing level. (Appendix A of Lobo et al. [11] contains a detailed discussion of how the problem instances are randomly generated.) Using randomly generated problems in this manner allows us to make sharper comparisons between the Virtual Factory’s performance using allocation \( \vartheta^{\text{HSP}} \) versus its performance using allocation \( \vartheta^* \); moreover, this performance comparison extends to the full range of staffing levels [21].

The maximum time allowed for searching was 60 s. In the real world, the allocation of workers in the job shop would typically occur no more frequently than once per shift; thus it was considered reasonable to spend up to one minute on searching for allocation \( \vartheta^{\text{HSP}} \). All experimentation was performed on computers that had an Intel Xeon W3520 processor running at 2.67 GHz with 4 GB of RAM.

5.2. Evaluation of HSP

For a given problem instance, if allocation \( \vartheta^* \) does not satisfy either of the optimality criteria, then the value \( V^{\text{F,VS}} \) represents the smallest \( l_{\text{max}} \) value that can be attained using the Virtual Factory to evaluate all feasible allocations. As a result, the quality of allocation \( \vartheta^{\text{HSP}} \) was measured by comparing \( V^{\text{F,VS}} \) with \( V^{\text{F,VS}} \), and not with \( L^{\text{P-L}} \), the lower bound on \( l_{\text{max}} \). The performance of allocation \( \vartheta^{\text{HSP}} \) relative to the VF-best allocation is defined as the difference \( V^{\text{F,VS}} - V^{\text{F,VS}} \). The average difference \( V^{\text{F,VS}} - V^{\text{F,VS}} \) is also given to illustrate the improvement in the average \( l_{\text{max}} \) value that can result from the use of HSP.

For a given class of DRC job shop scheduling problems, we can quantify the improvement in the average VF-value delivered by HSP using a one-sided skewness-adjusted 99% confidence interval for the expected value of the paired difference \( V^{\text{F,VS}} - V^{\text{F,VS}} \) that results from applying LBSA and HSP to the same randomly generated problem instance (from the given problem class) as explained in the fourth paragraph of Section 5.1. If the confidence interval for the mean difference excluded the value 0, then we concluded that using HSP yielded a statistically significant improvement in the average VF-value compared with the alternative strategy of merely using allocation \( \vartheta^* \). Section A2 of the Online Supplement provides a detailed procedure for calculating skewness-adjusted confidence intervals for the expected value of paired differences of the form \( D = V^{\text{F,VS}} - V^{\text{F,VS}} \). Owing to the limited space available, only selected skewness-adjusted confidence intervals are presented in this article. The reader is referred to Section A3 of the Online Supplement which contains all skewness-adjusted confidence intervals for the expected differences of the form \( E[V^{\text{F,VS}} - V^{\text{F,VS}}] \).

If allocation \( \vartheta^* \) satisfies either of the optimality criteria of Section 3, then it is not necessary to apply HSP to the problem at hand. Thus, for a given problem type using a particular staffing level, not all of the \( Q = 200 \) randomly generated problems required the application of HSP. Because of this, for a given type of job shop, staffing level, and due-date range, the average values of the differences \( V^{\text{F,VS}} - V^{\text{F,VS}} \) and \( V^{\text{F,VS}} - V^{\text{F,VS}} \) are necessarily averaged computed over 200 observations. In the tables containing the skewness-adjusted 99% confidence intervals for the mean of the paired differences, \( Q' \) is the number of randomly generated problems (out of 200) for which allocation \( \vartheta^* \) did not satisfy the hypotheses of either Theorem 1 or Theorem 2.
the random sample \( \{D_i : i = 1, \ldots, Q'\} \) of observed differences, we computed the sample mean \( \overline{D} \), the sample variance \( S_D^2 \), and the sample skewness \( \overline{Sk_D} \). Formulas for calculating each of these statistics can be found in Section A2 of the Online Supplement.

5.2.1. Symmetric job shop
For the symmetric job shop, Fig. 1 shows the average performance of allocations \( \beta^{\text{HSP}} \) and \( \beta^* \) when each allocation was compared with allocation \( \beta^{\text{VFB}} \). Tables 3 and 4 contain one-sided skewness-adjusted 99% confidence intervals for the mean of the paired difference \( VF_{\beta^*} - VF_{\beta^{\text{VFB}}} \) when the staffing levels were 60% and 70%, respectively. All confidence intervals (see Section A3 of the Online Supplement) indicated that for all staffing levels in a symmetric job shop, there was statistically significant improvement (i.e., a reduction) in the average value of \( L_{\text{max}} \) as a result of using the allocation \( \beta^{\text{HSP}} \) delivered by HSP compared with the alternative of simply using allocation \( \beta^{\text{VFB}} \).

5.2.2. Asymmetric job shop
For the asymmetric job shop, Fig. 2 shows the average performance of allocation \( \beta^* \) and allocation \( \beta^{\text{HSP}} \) when compared with allocation \( \beta^{\text{VFB}} \). The one-sided skewness-adjusted 99% confidence intervals for the mean of the paired difference \( VF_{\beta^*} - VF_{\beta^{\text{VFB}}} \) in the case of 80% and 90% staffing are given in Tables 5 and 6, respectively. All
confidence intervals (see Section A3 of the Online Supplement) indicated that for all staffing levels in an asymmetric job shop, there was statistically significant improvement (reduction) in the average value of $L_{\text{max}}$ as a result of using the allocation $W_{\text{HSP}}$ delivered by HSP compared with the alternative of simply using allocation $W_{\text{n}}$.

5.2.3. Overall results

The results indicated that HSP was effective for all classes of DRC job shop scheduling problems considered. This effectiveness was most apparent in the asymmetric job shop with 60% staffing, and the symmetric job shop with 60% and 70% staffing, where a VF-best allocation was found nearly 100% of the time; and in a large percentage of such problem instances, we also observed that LBSA terminated because Condition 2 was satisfied (see Table 8 of Lobo et al. [11]). HSP was also very effective in the asymmetric job shop with 70% staffing, where the reduction in the average $L_{\text{max}}$ value was approximately 2.30 average operation processing times. The effectiveness of HSP appeared to diminish slightly as the staffing level increased, and this trend was most noticeable in the symmetric job shop.

The effectiveness of HSP and the trend can be explained by considering the following observations (fully explained in Section 5.2.3 of Lobo et al. [11]):

1. If LBSA terminated because Condition 2 was satisfied, then the allocations $\bar{\eta}$ surrounding allocation $\tilde{\eta}$ satisfied $\bar{L}_{\eta} \geq L_{\text{max}}$, and usually satisfied $L_{\eta} > L_{\text{max}}$. 

**Table 5**

<table>
<thead>
<tr>
<th>Due-date range</th>
<th>$Q$</th>
<th>$D$</th>
<th>$S_0$</th>
<th>$\bar{S}_0$</th>
<th>99% Confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>1</td>
<td>0.000</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>600</td>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1000</td>
<td>15</td>
<td>0.107</td>
<td>0.205</td>
<td>1.814</td>
<td>(0.006, $\infty$)</td>
</tr>
<tr>
<td>1400</td>
<td>25</td>
<td>0.051</td>
<td>0.131</td>
<td>2.847</td>
<td>(0.005, $\infty$)</td>
</tr>
<tr>
<td>1800</td>
<td>27</td>
<td>0.204</td>
<td>0.354</td>
<td>1.975</td>
<td>(0.074, $\infty$)</td>
</tr>
<tr>
<td>2200</td>
<td>38</td>
<td>0.182</td>
<td>0.302</td>
<td>1.969</td>
<td>(0.087, $\infty$)</td>
</tr>
<tr>
<td>2600</td>
<td>49</td>
<td>0.167</td>
<td>0.257</td>
<td>2.006</td>
<td>(0.095, $\infty$)</td>
</tr>
<tr>
<td>3000</td>
<td>60</td>
<td>0.149</td>
<td>0.186</td>
<td>1.470</td>
<td>(0.074, $\infty$)</td>
</tr>
</tbody>
</table>

**Table 6**

<table>
<thead>
<tr>
<th>Due-date range</th>
<th>$Q$</th>
<th>$D$</th>
<th>$S_0$</th>
<th>$\bar{S}_0$</th>
<th>99% Confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>1</td>
<td>0.000</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>600</td>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1000</td>
<td>16</td>
<td>0.155</td>
<td>0.232</td>
<td>1.042</td>
<td>(0.031, $\infty$)</td>
</tr>
<tr>
<td>1400</td>
<td>26</td>
<td>0.060</td>
<td>0.130</td>
<td>2.564</td>
<td>(0.014, $\infty$)</td>
</tr>
<tr>
<td>1800</td>
<td>27</td>
<td>0.172</td>
<td>0.280</td>
<td>2.067</td>
<td>(0.069, $\infty$)</td>
</tr>
<tr>
<td>2200</td>
<td>36</td>
<td>0.172</td>
<td>0.270</td>
<td>1.976</td>
<td>(0.085, $\infty$)</td>
</tr>
<tr>
<td>2600</td>
<td>48</td>
<td>0.120</td>
<td>0.238</td>
<td>2.608</td>
<td>(0.056, $\infty$)</td>
</tr>
<tr>
<td>3000</td>
<td>60</td>
<td>0.186</td>
<td>0.219</td>
<td>1.180</td>
<td>(0.126, $\infty$)</td>
</tr>
</tbody>
</table>

Fig. 2. Average deviation of $V_{\text{HSP}}$ vs. $V_{\text{VF}}$ from $V_{\text{VF NSA}}$, asymmetric job shop.
2. If LBSA terminated because Condition 1 was satisfied, then there were a large number of allocations \( \mathcal{A} \) surrounding allocation \( \mathcal{A}^* \) for which \( \text{LB}_3 > \text{LB}_{\mathcal{A}} \).
3. As the staffing level increased, termination Condition 2 of LBSA occurred less frequently.

As described in Section 4.1, the partial enumeration strategy uses the initial requirement \( \text{LB}_3 < \text{VF}_{\mathcal{A}} \) to decide whether an allocation \( \mathcal{A} \) merits evaluation with the Virtual Factory. In light of the first observation, when Condition 2 was satisfied, there were, in general, relatively few allocations satisfying this initial requirement. In light of the second observation, when Condition 1 was satisfied, there were, in general, a large number of allocations satisfying this initial requirement. Therefore when Condition 2 was satisfied, the majority of allocations searched by the partial enumeration strategy were also searched by HSP. However, this was not the case when Condition 1 was satisfied. Combined with the third observation, this explains why HSP was most effective when the staffing level was lower, and why HSP’s effectiveness diminished as the staffing level increased.

If allocation \( \mathcal{A}^* \) does not satisfy the optimality criterion of Theorem 1 (i.e., if \( \text{VF}_{\mathcal{A}} \neq \text{VF}_{\mathcal{A}}^* \), then the value \( \text{VF}_{\mathcal{A}} \) is an upper bound on \( L_{\text{max}} \) for the \( J_{\text{max}}/W_{\text{max}} \) DRC job shop scheduling problem. Given that allocation \( \mathcal{A}^* \) did not satisfy either optimality criteria presented in Section 3, and allocation \( \mathcal{A}^* \) does not satisfy the optimality criterion of Theorem 1, it is important to have a way of assessing the quality of allocation \( \mathcal{A}^* \), and in turn, the relative quality of the upper bound on \( L_{\text{max}} \). The following section provides a probabilistic approach to making such an assessment.

6. A probabilistic approach to finding a VF-best allocation

If allocation \( \mathcal{A}^* \) does not satisfy either of the optimality criteria outlined in Section 3, and if allocation \( \mathcal{A}^* \) does not satisfy the optimality criterion of Theorem 1, then the following question arises: is the allocation \( \mathcal{A}^* \) delivered by HSP in fact a VF-best allocation? Enumeration of the allocation search space is guaranteed to yield a VF-best allocation, while heuristics (such as HSP) are not guaranteed to yield even a VF-best allocation. Therefore in the context of a specific DRC job shop scheduling problem (subsequently called the “designated problem”), it is important to assess the quality of an allocation \( \mathcal{A}^* \) by comparing its relative performance to the VF-best allocation. In particular each simulated problem is identical to the designated problem in the following key respects:

- It has the same number of jobs;
- It has the same number of machines and the same number of machine groups;
- It has the same pattern of symmetric or asymmetric loading of the machine groups;
- It has the same level of staffing—i.e., the ratio of the number of workers to the number of machines expressed as a percentage between 0% and 100%; and
- It has the same due-date range.

On the other hand for each simulated problem, the following characteristics of that problem are randomly sampled from appropriate probability distributions:

- the due-date of each job;
- the number of operations required to complete each job;
- the route of each job through the job shop (i.e., the ordered sequence of machines that the job must visit); and
- the processing time of each operation.

After generating \( Q \) simulated problems based on the designated problem, we restrict our attention to only the \( Q \) simulated problems in which the LBSA-delivered allocation \( \mathcal{A}^* \) did not satisfy either of the optimality criteria presented in Section 3, where of course \( Q \geq 1 \). The performance of an allocation \( \mathcal{A} \) relative to the lower bound on \( L_{\text{max}} \), denoted by \( \text{LB}_{\mathcal{A}} \), is defined as the difference between the following: (a) the value of \( L_{\text{max}} \) for the schedule generated by the Virtual Factory using allocation \( \mathcal{A} \); and (b) the lower bound on \( L_{\text{max}} \) for the LBSA-generated allocation \( \mathcal{A}^* \). Therefore we take

\[
\text{PLB}(\mathcal{A}) = \text{VF}_{\mathcal{A}} - \text{LB}_{\mathcal{A}}. \tag{10}
\]

For the \( i \)-th simulated problem \((i = 1, \ldots, Q)\) whose corresponding LBSA-delivered allocation \( \mathcal{A}^* \) does not satisfy the optimality criteria of Section 3, we define a VF-best allocation using the partial enumeration strategy presented in Section 4.1. A histogram can be constructed from the resulting data set comprising the \( Q \) values of \( \text{PLB}(\mathcal{A}^*) \), that is

\[
\{\text{PLB}(\mathcal{A}^*_i) : i = 1, \ldots, Q\}. \tag{11}
\]

The histogram’s horizontal axis, which represents the range of possible values of the random variable \( \text{PLB}(\mathcal{A}^*) \), is partitioned into equal-width bins (class intervals) of \( \text{PLB}(\mathcal{A}^*) \) values, where Scott’s rule [22] is used to determine the bin width and the number of bins in the histogram. The histogram’s vertical axis represents the likelihood with which \( \text{PLB}(\mathcal{A}^*) \) values fall in the corresponding bins; therefore for each bin, the area of the rectangle sitting over that bin is equal to the relative frequency with which the simulation-generated \( \text{PLB}(\mathcal{A}^*) \) values fall in that bin. Sections 6.1.2 and 6.3 that follow both contain examples of the histogram and the associated empirical cumulative distribution function (cdf) describing the data set (11) arising from a set of simulated problems that are based on a designated DRC job shop scheduling problem. An extensive collection of the aforementioned graphs can also be found in Lobo et al. [23].

Remark 7. In practice even for a medium-size DRC job shop scheduling problem and for a moderate value of \( Q \), it can be computationally prohibitive to determine a VF-best allocation.
for each simulated problem using the partial enumeration strategy presented in Section 4.1. We are currently developing computationally practical methods to generate random samples from a sufficiently close approximation to the desired distribution of PLB(δ). Although in this section we use the “ideal” approach to generating the data set (11) based on the partial enumeration strategy for finding a VF-best allocation, this ideal approach is intended mainly to demonstrate the potential of our method for estimating the probability (or the user’s degree of confidence) that a specific allocation δ delivered by HSP for a designated DRC job shop scheduling problem is a VF-best allocation for that problem.

6.1.2. A theoretical probability distribution describing PLB(δVFB)

Given a designated DRC job shop scheduling problem together with a specific allocation δ of workers to machines for that problem (where δ is perhaps the allocation δHSP delivered by HSP) such that δ does not satisfy the optimality criteria of Section 3, we seek to estimate our level of confidence that δ is a VF-best allocation for the designated problem. If δ is in fact a VF-best allocation, then we may regard the difference PLB(δ) = VF − LB, defined by Eq. (10) as another observation from the population of differences of the form

\[ PLB(\delta_{VFB}) = V_{FB} - L_{B}, \]

that encompasses all possible simulated problems based on the designated problem; and we will estimate the cdf of this population using the simulation-generated random sample (11). In the spirit of statistical hypothesis testing, we may therefore test the null hypothesis that δ is a VF-best allocation for the designated problem by estimating the probability that for a simulated problem based on the designated problem, the associated random variable PLB(δ) defined by Eq. (10) will be greater than the particular (fixed) value PLB(δ) for the designated problem. This upper tail probability can be viewed as the p-value (significance probability) for the test of the null hypothesis that δ is a VF-best allocation for the designated problem [24, pp. 221–223]. From a different perspective, we can interpret this upper tail probability as our level of confidence that δ is a VF-best allocation for the designated problem. For example, if approximately 95% of the observations in the data set (11) are larger than PLB(δ) for the designated problem, then we can conclude that allocation δ is better than approximately 95% of the VF-best allocations for simulated problems based on the designated problem; and thus we can be 95% confident that allocation δ is in fact a VF-best allocation for the designated problem.

6.2. Fitting a theoretical distribution to random samples of PLB(δVFB)

To facilitate their use in practice, we seek to approximate the histogram and the empirical cdf based on the data set (11) for a given problem class by fitting an appropriate standard probability distribution to that data set. We used the Stat::Fit software [25] for this purpose, which provided us with the following graphs:

- A graph of the histogram of the data set (11) superimposed on the fitted probability density function (pdf); and
- A graph of the empirical cdf of the data set (11) superimposed on the fitted cdf

In addition, a p-value for the chi-squared goodness-of-fit test was provided for each fitted distribution.

Because Q’ varied substantially across the different problem classes, care was taken when using the p-values as indicators of the goodness-of-fit. When Q’ is very large, “practically insignificant discrepancies between the empirical and theoretical distributions often appear statistically significant” [26]. On the other hand, very small values of Q’ often result in relatively large p-values because standard goodness-of-fit statistics have low power to distinguish between different distributions based on small samples. These considerations indicate that the p-value cannot be relied on as the sole goodness-of-fit criterion. The p-value from the chi-squared goodness-of-fit test and graphs of both the fitted pdf and the fitted cdf were used to decide which distribution best characterized a given data set.

If the data set (11) could be adequately modeled by a standard continuous distribution, then we fitted the associated pdf to the data set as detailed in Section 6.2.1 below. However, in some situations, the data set (11) exhibited a substantial percentage of observations at zero. In those situations, we used a mixed distribution with nonzero probability mass at the origin and with a continuous right-hand tail having a standard functional form as described in Section 6.2.2 below. In other situations we were forced to use a discrete probability mass function (pmf) to describe the data set (11) as explained in Section 6.2.3. The latter situation arose in cases where there were relatively few distinct nonzero values of PLB(δVFB) in the data set. In the fits that follow, the number of randomly generated problem instances for each class of DRC job shop scheduling problem is Q = 500.

6.2.1. Continuous distributions

For the data set (11) associated with each class of DRC job shop scheduling problems, Stat::Fit was initially used to seek the best-fitting continuous distribution. When adequate fits were obtained, the following distributions were used:

- the generalized Beta distribution, denoted as Beta(x, β, a, b), where x and β are the two shape parameters, a is the lower limit, and b is the upper limit;
- the shifted Gamma distribution, denoted as Gamma(x, β, a), where x is the shape parameter, β is the scale parameter, and a is the lower limit;
- the bounded Johnson distribution, denoted as Johnson-SB(γ, δ, λ, ξ), where γ and δ are shape parameters, λ is the scale parameter representing the range of possible values of the corresponding random variable, and ξ is the location parameter representing the lower limit of the distribution [26];
- the unbounded Johnson distribution, denoted as Johnson-SU(γ, δ, λ, ξ), where γ and δ are shape parameters, λ is the scale parameter (but not the range of possible values of the corresponding random variable, which is infinite in both directions), and ξ is a scale parameter (but not the lower limit of the distribution, which is −∞); see Kuhl et al. [26];
- the shifted Lognormal distribution, denoted as Lognormal(μ, σ, a), where μ and σ are the mean and standard deviation of the corresponding random variable respectively, and a is the lower limit; and
- the shifted (three-parameter) Weibull distribution, denoted as Weibull(x, λ, a), where x is the shape parameter, λ is the scale parameter, and a is the lower limit.

Fig. 3 depicts four examples of continuous distributions that were fitted to data sets of the form (11). Similar graphs and tables that summarize the fits for all problem classes fitted using a continuous distribution can be found in Lobo et al. [23]. In the corresponding graphs for data sets fitted with a continuous distribution, the legend “Datapoints” identifies the size of the data set, Q’. Both the graphical evidence and the p-values for the
goodness-of-fit tests indicated that in every problem class for which a continuous distribution was used to approximate the data set, the resulting model was adequate.

6.2.2. Mixed distributions

If there was a nonnegligible probability mass at $\text{PLB}(\omega_{VFB}) = 0$, then it was not possible to fit the data set with a standard

Fig. 3. Probability distribution fitting, continuous distribution fits.
continuous distribution. In these cases, the data sets were fitted using a mixed cdf of the form
\[ F(x) = p_0 F_0(x) + \left(1 - p_0\right) F_c(x) \]
for \(-\infty < x < \infty\),

where: (a) \(p_0\) is the proportion of observations of \(\text{PLB}(\text{VFB})\) that are equal to zero; (b) \(F_0(x)\) is the cdf of the degenerate distribution with unit probability mass at the origin so that
\[ F_0(x) = \begin{cases} 0 & \text{for } x < 0, \\ 1 & \text{for } x \geq 0; \end{cases} \]
and (c) \(F_c(x)\) is the conditional cdf of \(\text{PLB}(\text{VFB})\) given that \(\text{PLB}(\text{VFB}) > 0\), where we may take \(F_c(x)\) to be the cdf of any of the distributions listed in Section 6.2.1.

Two examples of problems fitted with a mixed distribution are: (a) the asymmetric job shop with 70% staffing and a due-date range of 1800; and (b) the symmetric job shop with 90% staffing and a due-date range of 2600. Fig. 4 displays the continuous distribution that provided the “best fit” to the associated subsample consisting of the nonzero values of \(\text{PLB}(\text{VFB})\) for each of the problem classes. The graphs in Fig. 4 also show \(Q\left(1 - p_0\right)\), the size of the associated subsample (labeled “Datapoints”), and the \(p\)-value for the chi-squared goodness-of-fit test. Tables that summarize the “best fits” for all problem classes fitted using a mixed distribution, together with the associated graphical fits can be found in Lobo et al. [23]. Both the graphical evidence and the \(p\)-values for the goodness-of-fit tests indicated that in every problem class for which a mixed distribution was used to approximate the data set, the resulting model was adequate.

6.2.3. Discrete distributions
For a number of problem types with given staffing levels (e.g., the asymmetric job shop with a due-date range of 2200 and 90% staffing), there were so few \(\text{PLB}(\text{VFB})\) values distinct from zero in the data set that a continuous distribution could not be reliably fit to the remaining nonzero values. In this case, a discrete pmf was determined for the data set. See Lobo et al. [23] for an example of a discrete pmf fitting.

6.2.4. Fitting a distribution in general
The overall experimental design was composed of 64 different classes of DRC job shop scheduling problems (i.e., 64 different combinations of job shop type, staffing level, and due-date range). For each problem class, the data set of \(\text{PLB}(\text{VFB})\) values was constructed as described in Section 6.1 with \(Q = 500\). Due to the limited space available, the graphs pertaining to the 64 different combinations cannot be presented in this article. The full set of graphs, along with an in depth description and critique of the fitting methodology, can be found in Lobo et al. [23].

More than 10% of the \(\text{PLB}(\text{VFB})\) values were equal to zero for each of the following cases: (a) the symmetric job shop with 90% staffing and a due-date range of 200 through 2600; (b) the asymmetric job shop with 70% staffing; (c) the asymmetric job shop with 80% staffing and a due-date range of 2600 and 3000; and (d) the asymmetric job shop with 90% staffing and a due-date range of 2600 and 3000. In these cases, a conditional pdf was fitted to the data set composed of the nonzero \(\text{PLB}(\text{VFB})\) values. Less than 50 nonzero \(\text{PLB}(\text{VFB})\) values were observed in each of the following cases.

Fig. 4. Probability distribution fitting, part \(F_c(\cdot)\) of the mixed distribution fits.
cases: (i) the asymmetric job shop with 80% staffing and a due-date range of 200 through 2200; and (ii) the asymmetric job shop with 90% staffing and a due-date range of 200 through 2200. In cases (i) and (ii), we simply used the observed values in each data set to define a discrete probability distribution.

We found that, in general, the generalized Beta distribution, the shifted Gamma distribution, the shifted Lognormal distribution, and the shifted Weibull distribution could be used to characterize the majority of the data sets. The exception to this was the symmetric job shop with 80% staffing, where the unbounded Johnson distribution approximated the left-hand tail of the associated data sets substantially better than any of the other standard continuous distributions. (Kuhl et al. [26] discuss several diverse engineering applications in which unbounded Johnson distributions yield superior fits to nonstandard tail behavior.) In the symmetric job shop with 80% staffing and a due-date range of 3000 the small p-value was not a concern when the value of \( Q \) and the graphs of the fitted pdf and cdf (see Fig. 3) were taken into account.

The practical applicability of our distribution-fitting approach is clearly demonstrated by the diversity of DRC job shop scheduling problems for which the associated data set (11) can be adequately modeled by these six different probability distributions in a straightforward manner.

6.3. Using the theoretical probability distribution

This section illustrates the use of the theoretical probability distribution through two examples, and demonstrates the improvement in allocation quality that is achievable through use of HSP.

A random problem was generated for an asymmetric job shop with 60% staffing and a due-date range of 1400. The fitted pdf corresponding to such a combination of job shop type, staffing level, and due-date range is \( f_{\text{asym}}(x) \sim \text{Weibull}(1.36, 55.42, 11.49) \) (see Fig. 5). Allocation \( \theta^* \) did not satisfy either of the optimality criteria presented in Section 3, where \( LB_{\theta^*} = 3162, VF_{\theta^*} = 3313 \), and therefore \( PLB(\theta^*) = 151 \).

From this we know that

\[
\int_0^{PLB(\theta^*)} f_{\text{asym}}(x) \, dx = \Pr(PLB(\theta^{\text{VF*}}) \leq 151)
\]

\[
= 1 - \exp \left( -\frac{151 - 11.49}{55.42} \right)^{1.36}
\]

\[
= 0.9701,
\]

that is, we are only 2.99% confident allocation \( \theta^* \) is a VF-best allocation. After using HSP, we found that \( PLB(\theta^{\text{HSP}}) = 44 \), and thus we can conclude that

\[
\Pr(PLB(\theta^{\text{VF*}}) \leq PLB(\theta^{\text{HSP}})) = \Pr(PLB(\theta^{\text{VF*}}) \leq 44) = 0.3838,
\]

indicating we are 61.62% confident that allocation \( \theta^{\text{HSP}} \) is a VF-best allocation.

A random problem was generated for a symmetric job shop with 80% staffing and a due-date range of 2200. The fitted pdf corresponding to such a combination of job shop type, staffing level, and due-date range is \( f_{\text{sym}}(x) \sim \text{Johnson-SU}(-0.45, 4.40, 116.97, 75.00) \) (see Fig. 6). Again, allocation \( \theta^* \) did not satisfy either of the optimality criteria presented in Section 3, where \( LB_{\theta^*} = 1146, VF_{\theta^*} = 1337 \), and therefore \( PLB(\theta^*) = 191 \). We find that

\[
\int_0^{PLB(\theta^*)} f_{\text{sym}}(x) \, dx = \Pr(PLB(\theta^{\text{VF*}}) \leq 191)
\]

\[
= \Phi \left( -0.45 + 4.40 \ln \left( \frac{191 - 75.0}{116.97} + \sqrt{\left( \frac{191 - 75.0}{116.97} \right)^2 + 1} \right) \right)
\]

\[
= 0.9997,
\]

(where \( \Phi(\cdot) \) denotes the standard normal cdf), indicating we are only 0.03% confident that \( \theta^* \) is a VF-best allocation. After using HSP we found that \( PLB(\theta^{\text{HSP}}) = 74 \), from which we can conclude that

\[
\Pr(PLB(\theta^{\text{VF*}}) \leq PLB(\theta^{\text{HSP}})) = \Pr(PLB(\theta^{\text{VF*}}) \leq 74) = 0.313,
\]

indicating that we are 68.7% confident that \( \theta^{\text{HSP}} \) is a VF-best allocation.

Both examples illustrate that, when allocation \( \theta^* \) does not satisfy the optimality criteria developed in Section 3, HSP is able to find a significantly improved allocation \( \theta^{\text{HSP}} \). The theoretical probability distributions measure the quality of allocation \( \theta^{\text{HSP}} \) by quantifying the uncertainty introduced by a heuristic search method.

The analysis in this section was developed using the Virtual Factory as the heuristic scheduler. The theoretical basis for this analysis would be just as valid if another heuristic scheduler were used.

7. Conclusions and recommendations

In this article we formulate and justify optimality criteria by which the allocation \( \theta \) that yields the smallest value of \( LB_\theta \) can be declared optimal. In the case that allocation \( \theta^* \) fails to satisfy either of the optimality criteria, we develop three different

![Fig. 5. Probability distribution fitting, asymmetric job shop, 60% staffing, 1400 due-date range.](image-url)
heuristic strategies that search for a VF-best allocation—i.e., an allocation that enables the Virtual Factory to generate a schedule with the smallest possible value of $L_{\text{max}}$ that can be achieved by applying the Virtual Factory to the problem at hand. The experimental results indicated that the search strategies, on the whole, provided a statistically significant improvement over the $L_{\text{max}}$ value obtained when using the LBSA-delivered allocation $\hat{\gamma}$, together with the Virtual Factory, to generate a schedule for the job shop. We also present a simulation-based method for obtaining a probability distribution from which, for a designated DRC job shop scheduling problem, we can estimate our degree of confidence that a specific allocation $\hat{\gamma}$ is in fact a VF-best allocation for the designated problem. This probability distribution can be used to evaluate allocation quality, and quantify the uncertainty introduced through the use of a heuristic scheduler. As mentioned in Remark 7, rapid computational methods are required to obtain a simulation-based approximation to this distribution in practical problems of realistic size and complexity.

Developing and implementing such methods is the primary objective of our ongoing research. Finally, even if an optimal allocation is not found, the value $VF_{\text{opt}}$ forms an upper bound on $L_{\text{max}}$ for the $\frac{H_{\text{sym}}}{W_{\text{max}}}$ DRC job shop scheduling problem.

A promising direction for future work is to design a stopping criterion for heuristic search strategies that makes effective use of the probability distributions formulated in this article. A good candidate for such a stopping criterion would be LNSS, whose average execution time is the longest. Another direction for future work is to extend the current research to handle a DRC job shop in which workers have specific skills, and the workers may only be assigned to machine groups matching their skills. Although the latter system involves a more difficult worker-allocation problem, it is perhaps a more accurate representation of many DRC job shops encountered in practice. We have good reason to believe that the framework developed in this article can be applied to such DRC job shops.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cor.2013.02.008.

References


