Allocating Manpower to Minimize $L_{\text{max}}$ in a Job Shop
A Heuristic Search Procedure

Benjamin J. Lobo

Edward P. Fitts Department of Industrial & Systems Engineering,
North Carolina State University

May 24th, 2011

http://people.engr.ncsu.edu/bjlobo/

bjlobo@gmail.com
Job Shop Scheduling

- Determine an optimal schedule given a performance objective

- Active area of research for more than 60 years [Potts & Strusevich, 2009]

- Most job shop scheduling problems are NP-Hard [Lenstra et al., 1977]

- Basic assumption: machines form the only constraint

- In practice, workers form a constraint!
Dual Resource Constrained Systems

- Dual Resource Constrained (DRC) system — “One in which shop capacity may be constrained by machine and labor capacity or both” [Treleven & Elvers, 1985]

- Partition job shop into machine groups, assign workers to machine groups

- The worker allocation affects job shop performance

- Determine the allocation of workers in the job shop that allows $L_{\text{max}}$ to be minimized
The $N/M/L_{\text{max}}$ Problem

- Hodgson et al. [1998] consider the classic $N/M/L_{\text{max}}$ job shop scheduling problem: Find a schedule that minimizes $L_{\text{max}}$ in a job shop that has $N$ jobs, $M$ machines, and the release times of the jobs ($r_j$) are not all the same.


- Approach:
  1. Calculate a lower bound on $L_{\text{max}}$ for the job shop
  2. Generate a schedule heuristically
  3. Compare the actual $L_{\text{max}}$ value to the lower bound value to evaluate the schedule
Model Assumptions

New assumptions for the 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 it cannot process any jobs
- A machine takes one worker to operate it, and a worker cannot operate more than one machine at a time
- The allocation of workers is static for a given problem
- Every worker is fully cross-trained
Finding a Lower Bound on $L_{\text{max}}$ for the DRC Problem

- Adopted a similar approach to Hodgson et al. [1998]

- Given an allocation $\vartheta$:
  - Network flow approach: $\text{LB}_{mg}$ — the lower bound on $L_{\text{max}}$ for a machine group
  - Maximum over lower bound on $L_{\text{max}}$ for each machine group $\rightarrow$ lower bound on $L_{\text{max}}$ for the job shop ($\text{LB}_{\vartheta}$)

- Lower Bound Search Algorithm (LBSA) used to find allocation $\vartheta^*$

- **Allocation** $\vartheta^*$: tightest lower bound on $L_{\text{max}}$

- Evaluate allocation $\vartheta^*$ using the Virtual Factory
Optimality Conditions & Experimental Observations

- **Optimal** allocation: the schedule generated using this allocation will have the minimum possible value of $L_{\text{max}}$ for the $N/\{M, W\}_{js}/L_{\text{max}}$ job shop scheduling problem.

- Defined two optimality conditions.

- Experimentally observed: Allocation $\vartheta^*$ does not always satisfy optimality conditions.

- Enumeration of allocation search space allows us to find:
  - **VF–best** allocation: allocation that yields the VF-generated schedule with $\text{VF}_{\vartheta^*_{\text{VF}}} L_{\text{max}}$, the smallest $L_{\text{max}}$ value.

- Enumeration is computationally impractical.
Permuting an Allocation

- **Neighbor** of allocation $\vartheta$: has the same number of workers in all machine groups except that
  1. machine group $j$ has one less worker, and
  2. machine group $k$ has one more worker

**Permutation Strategy**

**A.** Identify:
   (i) Receiving machine group: one that “most needs” an additional worker
   (ii) Donating machine group: one that can “most afford” to lose a worker

**B.** Unassign a worker from the donating machine group, and reassign the worker to the receiving machine group
Alternatives to Enumeration

- Three different heuristic search strategies considered
  1. Local Neighborhood Search Strategy (LNSS)
  2. Queuing Time Search Strategy Version 1 (QSS1)
  3. Queuing Time Search Strategy Version 2 (QSS2)

- LNSS: uses $LB_{mg}$ values
- QSS1, QSS2: use queuing time of the $L_{max}$ job

**Heuristic Search Procedure (HSP)**
- LNSS $\rightarrow$ QSS1 $\rightarrow$ QSS2
- Maximum of 60 seconds of search time
Experimental Job Shop

- 1200 jobs, 10 machine groups each with 8 machines (total of 80 machines)
- Each job has Uniform(6, 10) operations, and visits the same machine group no more than 3 times
- Operation time is Uniform(1, 40)
- Overall job shop staffing levels of 60%, 70%, 80%, and 90% (e.g. 60% staffing = 48 workers for allocation)
- Symmetric and asymmetric job shop
- 200 replications at each job shop type, due-date range and staffing level combination
Experimental Results – Graphs

- Compare $L_{\text{max}}$ value of schedules using allocation $\vartheta^*$ or allocation $\vartheta^{\text{HSP}}$ to $L_{\text{max}}$ value of schedule using allocation $\vartheta^{\text{VFB}}$

- On the following graphs:
  - **Red bar**: average $\text{VF}_{\vartheta^*} - \text{VF}_{\vartheta^{\text{VFB}}}$ value
  - **Gray bar**: average $\text{VF}_{\vartheta^{\text{HSP}}} - \text{VF}_{\vartheta^{\text{VFB}}}$ value

- Shorter bar is better

- **Note**: asymmetric job shop, 60% staffing, HSP always finds a VF–best allocation except for a due-date range of 1000 and 2200
Average Deviation from $\text{VF}_{\text{\theta VFB}}$ vs. Due-date Range, Asymmetric Job Shop, 80 Machines, 60% Staffing

- Alloc. $\text{\theta}^*$
- Alloc. $\text{\theta}^{HSP}$
Average Deviation from $\text{VF}_{\theta_{VFB}}$ vs. Due-date Range, Asymmetric Job Shop, 80 Machines, 70% Staffing
Average Deviation from $VF_\theta^{VFB}$ vs. Due-date Range, Symmetric Job Shop, 80 Machines, 80% Staffing

Legend:
- Alloc. $\theta^*$
- Alloc. $\theta^{HSP}$

Due-date Range in Units of Time

VF$_{\theta}^{VFB}$ in Units of Avg. Oper. Proc. Time
Average Deviation from $VF_{\vartheta^{VFB}}$ vs. Due-date Range, Symmetric Job Shop, 80 Machines, 90% Staffing

<table>
<thead>
<tr>
<th>Due-date Range in Units of Time</th>
<th>Alloc. $\vartheta^*$</th>
<th>Alloc. $\vartheta^{HSP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formal Evaluation of Heuristic Search Procedure

- Estimate the expected value of $D = VF_{\vartheta^*} - VF_{\vartheta_{\text{HSP}}}$

- Constructed one-sided skewness-adjusted 99% confidence interval (see Willink [2005, 2007]) on $E[D]$

- Confidence interval formally demonstrates value of using the HSP (over just using allocation $\vartheta^*$)

- Heuristic Search Procedure (HSP) provides statistically significant improvement to average $L_{\max}$ value
99% Skewness-Adjusted Confidence Intervals

Table: Average Value of $D = VF_{\theta^*} - VF_{\theta_{HSP}}$ (Expressed in Units of Average Operation Processing Time), Asymmetric Job Shop, 70% Staffing

<table>
<thead>
<tr>
<th>Due-date Range</th>
<th>$Q'$</th>
<th>$\bar{D}$</th>
<th>$S_D$</th>
<th>$\widehat{Sk_D}$</th>
<th>99% Conf. Int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>88</td>
<td>2.173</td>
<td>2.643</td>
<td>2.968</td>
<td>$[1.635, \infty)$</td>
</tr>
<tr>
<td>600</td>
<td>109</td>
<td>2.500</td>
<td>2.668</td>
<td>1.108</td>
<td>$[1.945, \infty)$</td>
</tr>
<tr>
<td>1000</td>
<td>119</td>
<td>2.602</td>
<td>2.715</td>
<td>1.062</td>
<td>$[2.059, \infty)$</td>
</tr>
<tr>
<td>1400</td>
<td>130</td>
<td>2.426</td>
<td>2.518</td>
<td>1.300</td>
<td>$[1.950, \infty)$</td>
</tr>
<tr>
<td>1800</td>
<td>129</td>
<td>2.504</td>
<td>2.559</td>
<td>1.282</td>
<td>$[2.018, \infty)$</td>
</tr>
<tr>
<td>2200</td>
<td>135</td>
<td>2.502</td>
<td>2.594</td>
<td>1.446</td>
<td>$[2.025, \infty)$</td>
</tr>
<tr>
<td>2600</td>
<td>143</td>
<td>2.239</td>
<td>2.842</td>
<td>1.957</td>
<td>$[1.744, \infty)$</td>
</tr>
<tr>
<td>3000</td>
<td>146</td>
<td>2.288</td>
<td>2.927</td>
<td>1.982</td>
<td>$[1.784, \infty)$</td>
</tr>
</tbody>
</table>
Conclusions & Further Research

- Allocation $\vartheta^*$ provides the tightest lower bound on $L_{\max}$, but does not lead to schedule with the minimum $L_{\max}$ value.

- Enumeration is not computationally practical.

- The HSP provides a tradeoff between time spent searching and solution quality.

- HSP provided a statistically significant improvement to $L_{\max}$

Further research:

- Incorporate probabilistic stopping criterion into search strategies.
Any questions?

http://people.engr.ncsu.edu/bjlobo/

bjlobo@gmail.com
LB_{mg_i}(w_i): A Network Flow Formulation

- Want LB — a lower bound on $L_{\text{max}}$ for the job shop
- First find LB_{mg_i}(w_i) — a true lower bound on $L_{\text{max}}$ for machine group $i$, denoted by $mg_i$, given $w_i$ workers assigned to $mg_i$
- Use a network flow formulation
- Types of Nodes:
  - $s_i$: source node for all jobs in machine group $i$
  - $j_{u,h}$: job node for job $u$ on machine $h$
  - $t_{v,h}$: time period node for time period $v$ on machine $h$
  - $\omega_v$: worker availability node for time period $v$
  - $k_i$: sink node for machine group $i$
A Heuristic Search Procedure for Worker Allocation

The New DRC ($N/\{M, W\}_{js/L_{max}}$) Problem

Dashed arcs represent arcs to and from other machines in the $i^{th}$ machine group.
LB_{mg_i}(w_i): A Network Flow Formulation, cont’d.

- **Arc capacities and flow:**

<table>
<thead>
<tr>
<th>Arc</th>
<th>Capacity</th>
<th>Relevant Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>((s_i, j_u, h))</td>
<td>([p_{uh}, p_{uh}])</td>
<td>(u \in J_{ih}, \ h \in M_i)</td>
</tr>
<tr>
<td>((j_u, h, t_v, h))</td>
<td>([0, 1])</td>
<td>(u \in J_{ih}, \ v = \delta_h, \ldots, \tau_h, \ h \in M_i)</td>
</tr>
<tr>
<td>((t_v, h, \omega_v))</td>
<td>([0, 1])</td>
<td>(v = \delta_h, \ldots, \tau_h, \ h \in M_i)</td>
</tr>
<tr>
<td>((\omega_g, k_i))</td>
<td>([0, w_i])</td>
<td>(g = \psi, \ldots, T)</td>
</tr>
</tbody>
</table>

- Arc capacities ensure that:
  1. The entire job is processed
  2. At most one time unit of a job’s total processing time is completed in a given time period
Overview of One Iteration of the Constructive Algorithm

1. Generate the optimal preemptive schedule from time $t$ onwards for each machine not scheduled to operate on the last iteration.

2. At time $t$, schedule the $w_i$ machines with the $w_i$ largest $L_{max}$ values for operation.

- The maximum $L_{max}$ value over all the machines is the initial $y$ value.
Local Neighborhood Search Strategy

Type—α Allocations

- Ignore any machine group $k \in K$
- Unassign a worker from machine group $j_1$ and reassign it to machine group $j_2$ if:
  - Machine group $j_1$ has the smallest lower bound on $L_{\text{max}}$ value
  - Machine group $j_1$ has more than 1 worker
  - $LB_{mg_{j_1}} \leq LB_{\vartheta^*}$ after unassignment of the worker
  - Machine group $j_2$ has the largest lower bound on $L_{\text{max}}$ value
  - Machine group $j_2$ can receive a worker

- Each type—α allocation has same optimal lower bound on $L_{\text{max}}$ value — $LB_{\vartheta^*}$
Local Neighborhood Search Strategy, cont’d.

**Type−β Allocations**

- A neighbor allocation $\phi$ of an allocation $\vartheta$ is defined by:
  1. Unassigning a worker from machine group $j$, and
  2. Reassigning the worker to the currently selected constraining machine group $k$

- To create a type−β allocation:
  - For each neighbor allocation $\phi$, reassign the worker to any machine group other than $k$, to create allocation $\phi'$

- Note that $\text{LB}_\phi = \text{LB}_{\phi'}$
Queuing Time Search Strategy, v1

- $\text{Proc}_i$: Processing time of all operations in machine group $i$
- $L_{\text{max}}Q_i$: Estimated queuing time of $L_{\text{max}}$ job in machine group $i$
- At least 1 worker to each machine group. Additional workers assigned to machine group $i$:
  \[
  \min \left\{ \left\lfloor \frac{\text{Proc}_i + L_{\text{max}}Q_i}{\text{Proc}_{js} + L_{\text{max}}Q_{js}} \cdot (W - S) \right\rfloor, |\mathcal{M}_i| \right\} - 1
  \]

- Remaining workers allocated sequentially:
  \[
  \left( \frac{\text{Proc}_i + L_{\text{max}}Q_i}{\text{Proc}_{js} + L_{\text{max}}Q_{js}} \cdot (W - S) \right) - \left\lfloor \frac{\text{Proc}_i + L_{\text{max}}Q_i}{\text{Proc}_{js} + L_{\text{max}}Q_{js}} \cdot (W - S) \right\rfloor
  \]
Queuing Time Search Strategy, v2

- **Base allocation**: allocation with smallest $L_{\text{max}}$ value encountered so far

- **Receiving machine group**:
  - Largest $L_{\text{max}Q_i}$ value
  - Lies on $L_{\text{max}}$ job’s route
  - Can have a worker reassigned to it

- **Donating machine group**:
  - Smallest $\text{Proc}_i$ value
  - Does not lie on $L_{\text{max}}$ job’s route
  - Can have a worker unassigned from it