IISE ANNUAL CONFERENCE & EXPO 2017

Cooperative Workforce Planning Heuristic with Worker Learning and Forgetting, and Demands Constraints

PennState

Christian E. López B.
Department of Industrial and Manufacturing Engineering, The Pennsylvania State University.

OSU Oregon State University

Dr. David Nembhard
Department of Mechanical, Industrial and Manufacturing Engineering, The Oregon State University.

“People are still the greatest assets”

W. Vanderbloemen [2016]

Efficiency

Flexibility

FLOW TO LEARN

Decades of successes and failures show that proper lean leverages human ingenuity, not automation

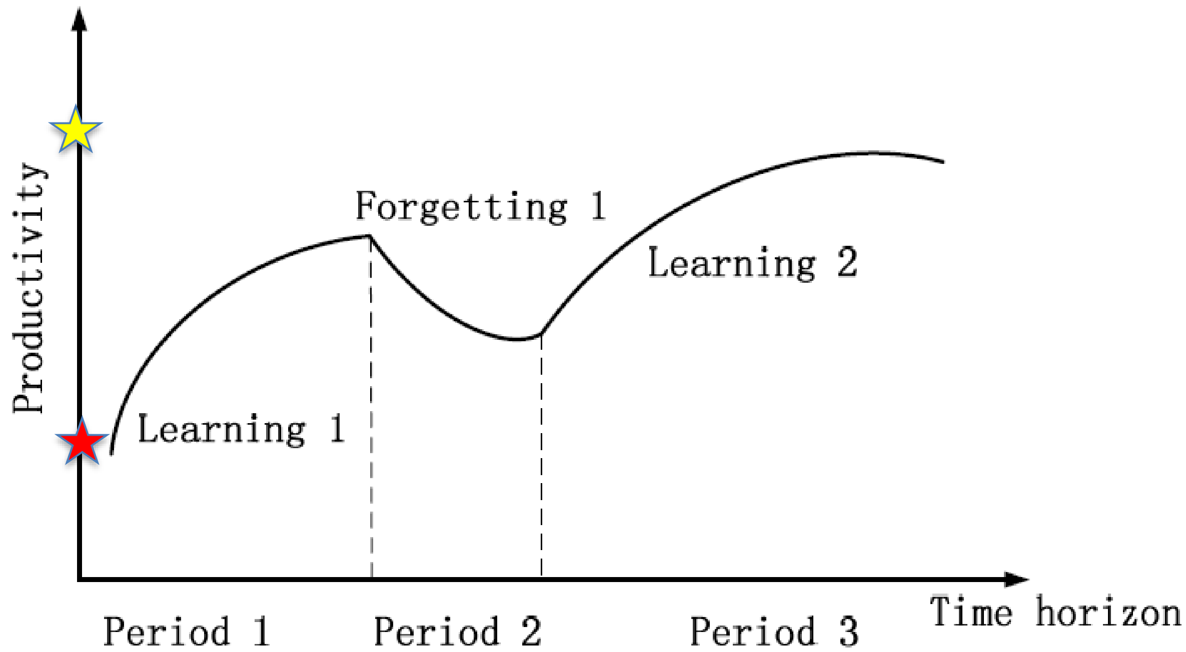
By Michael Ballé and Daniel Jones

ISE Magazine, May 2017, 9(5)

Workers Learning and Forgetting effects can impact productivity estimations.

Shafer et al. [2001]

*"The time needed to produce a single unit continuously decreases with the processing of additional units"**



Yang & Wang [2011], *Biskup [2008]

Learning and Forgetting increases the complexity of the scheduling problem

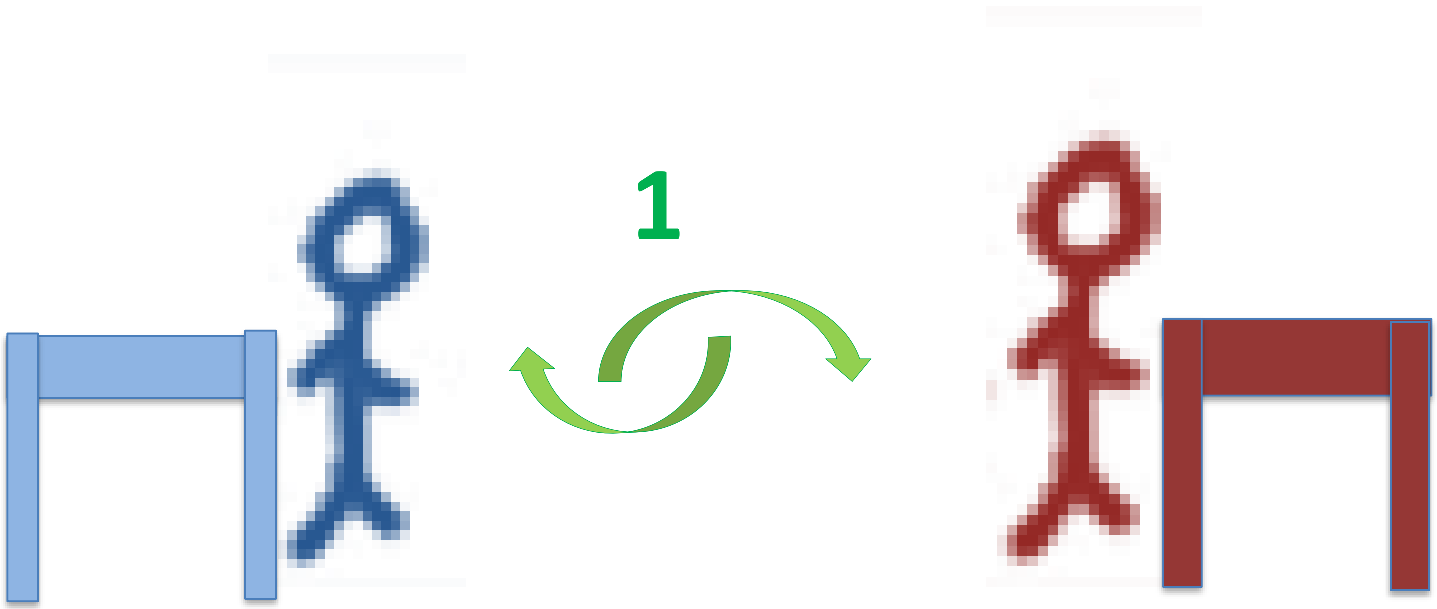
Shafer et al. [2001]

- Nembhard and Bentefouet (2012): ***Reformulation to MIP***
- Hewitt et al. (2015): ***Reformulation , Scaling Algorithm***
- Jin et al. (2016): ***Integer Programming Techniques***
- Wang et al. (2016): ***Branch-and-bound, Meta-heuristics***

Let’s take advantage of the optimal solution structure.

“Within parallel systems, when the demand is set for production during the entire time horizon and considering a differentiable non-decreasing model of performance, the maximum number of stints per worker per task is one”

Theorem 2 Nembhard and Bentefouet (2012)



Multi-period two parallel station system scheduling problem with L/F and demand constraints

Nembhard & Bentefouet [2012]

1. A worker's performance is a function of their skills and experience.
2. A one-to-one task-station relation.
3. No starvation or blocking.
4. The time horizon can be decomposed into same length periods.
5. At each period, a worker can only be assigned to a single task.
6. A one-to-one worker-task relation.

Mathematical formulation: Two workers, Two station systems

J	Set of tasks, j=1-2	T_{max}	Completion time of the last task to be finished.
I	Set of workers, i=1-2	$f_{i,j}$	The estimated productivity of worker i at task j.
T	Set of time periods in the horizon, t=1,…, T	$O_{i,j,t}$	The output of worker i performing task j during period t.
$x_{i,j,t}$	Binary variable that indicates whether task j is performed by worker i during time t.	D_j	The demand order size required by task j at time period T, measured in product units.

$$\text{Min } T_{max} \qquad (1)$$

$$t * x_{i,j,t} \le T_{max} \quad \forall i, \forall j, \forall t \qquad (2)$$

$$\sum_{i=1}^{2} x_{i,j,t} \le 1 \quad \forall j, \forall t \qquad (3)$$

$$\sum_{j=1}^{2} x_{i,j,t} \le 1 \quad \forall i, \forall t \qquad (4)$$

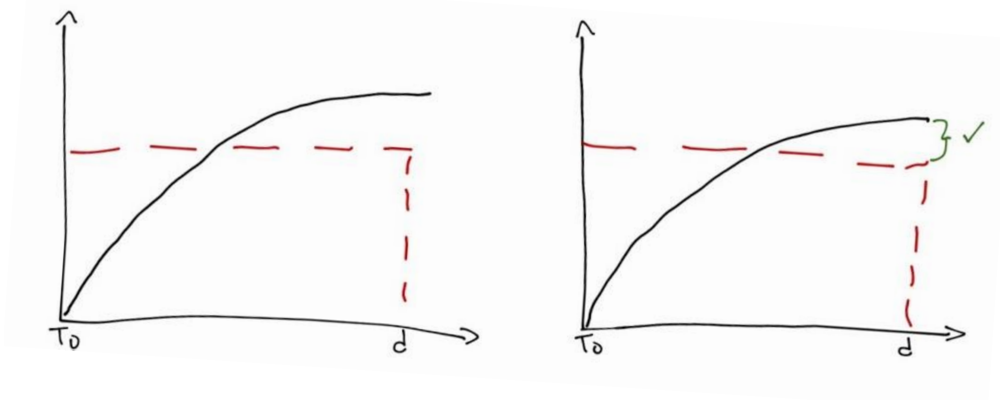
$$O_{i,j,t} \le f_{i,j}\left(\sum_{k=1}^{t} x_{i,j,k}\right) \times length\ of\ period \quad \forall i, \forall j, \forall t \qquad (5)$$

$$\sum_{i=1}^{2} \sum_{t=1}^{T} x_{i,j,t} * O_{i,j,t} \ge D_j \quad \forall j \qquad (6)$$

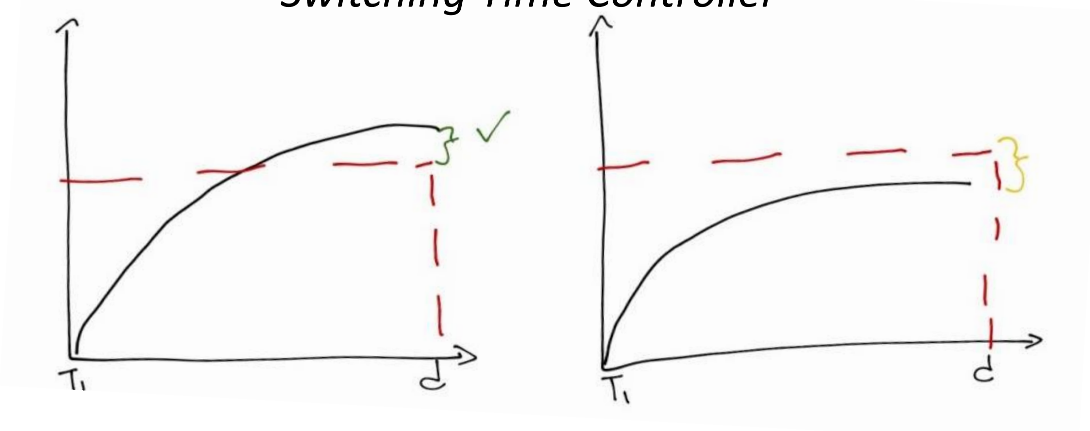
Cooperative Workforce Planning Heuristic

Iteration 1…

Arrival Time Controller



Switching Time Controller



Accumulative production [units]

Time Horizon [t1-T]

1) Select an initial worker assignment.

2) Checks if the current schedule meets the demand of both stations.

3) Generate new Schedule and test.

Design of Experiment

Learning function:

Three-parameter Hyperbolic Learning*

- $\boldsymbol{p_{i,j}}$ represented the initial productivity level of worker *i* at task *j*
- $\boldsymbol{r_{i,j}}$ the learning rate parameter of worker *i* at task *j*
- $\boldsymbol{k_{i,j}}$ the steady state production rate of worker *i* at task *j.*

$$f_{i,j}(t) = k_{i,j} \frac{t + p_{i,j}}{t + p_{i,j} + r_{i,j}}$$

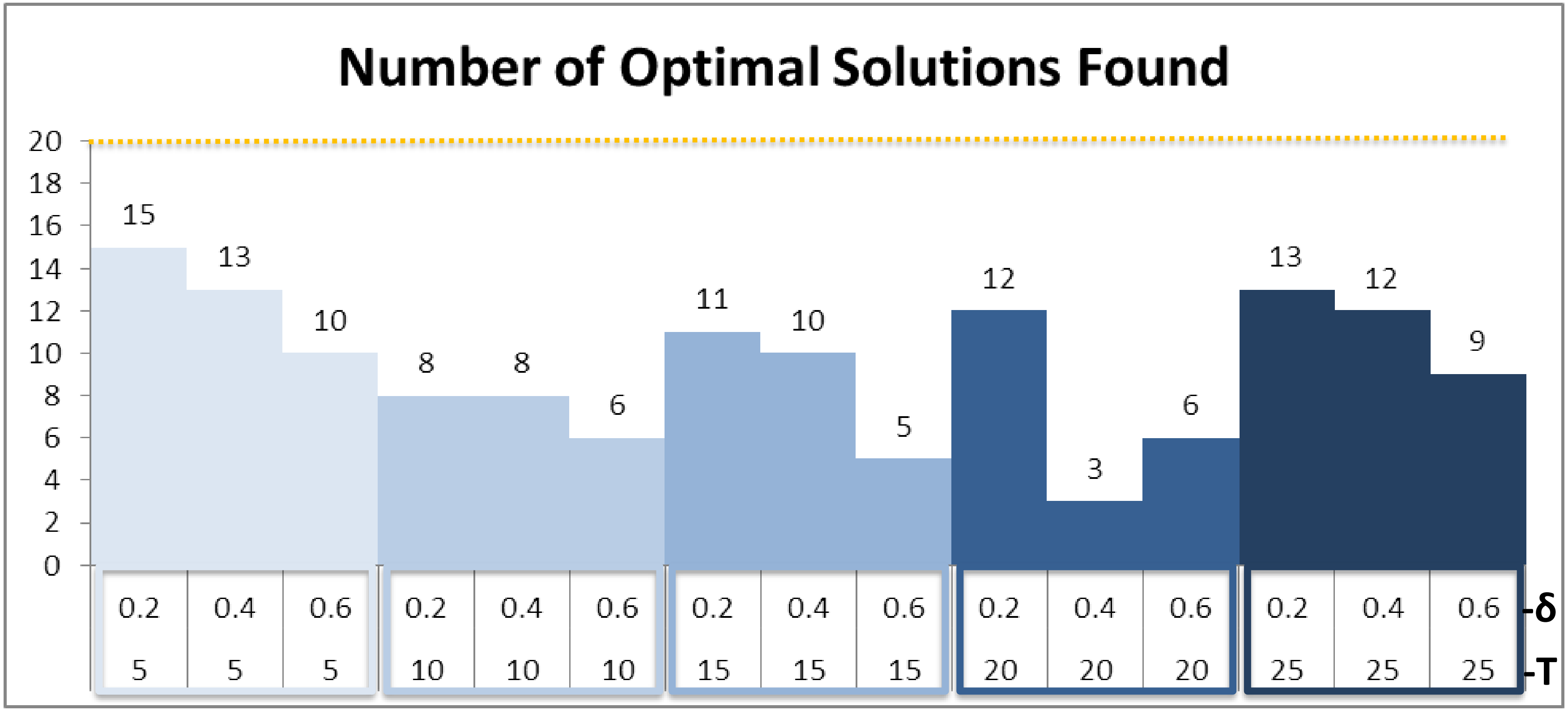
Experimental conditions:

1) **Time horizon periods** (T) = 5,10,15,20,25
2) **Demand order factor** (δ)= 0.2, 0.4, 0.6

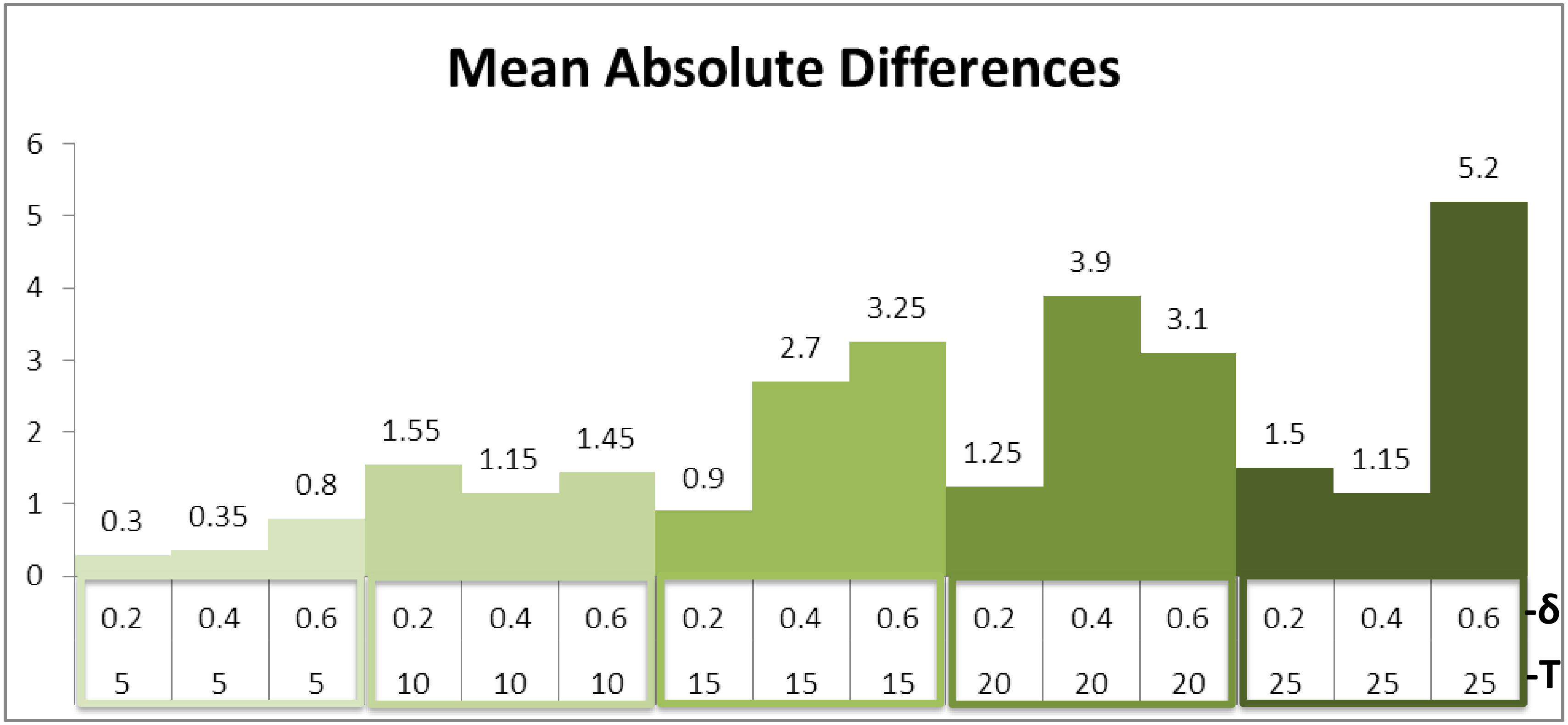
Demand order size sampled from Uniform Dist. :[(20*Tδ/ 2*), (20*Tδ*)]

*Nembhard and Uzumeri [2000]

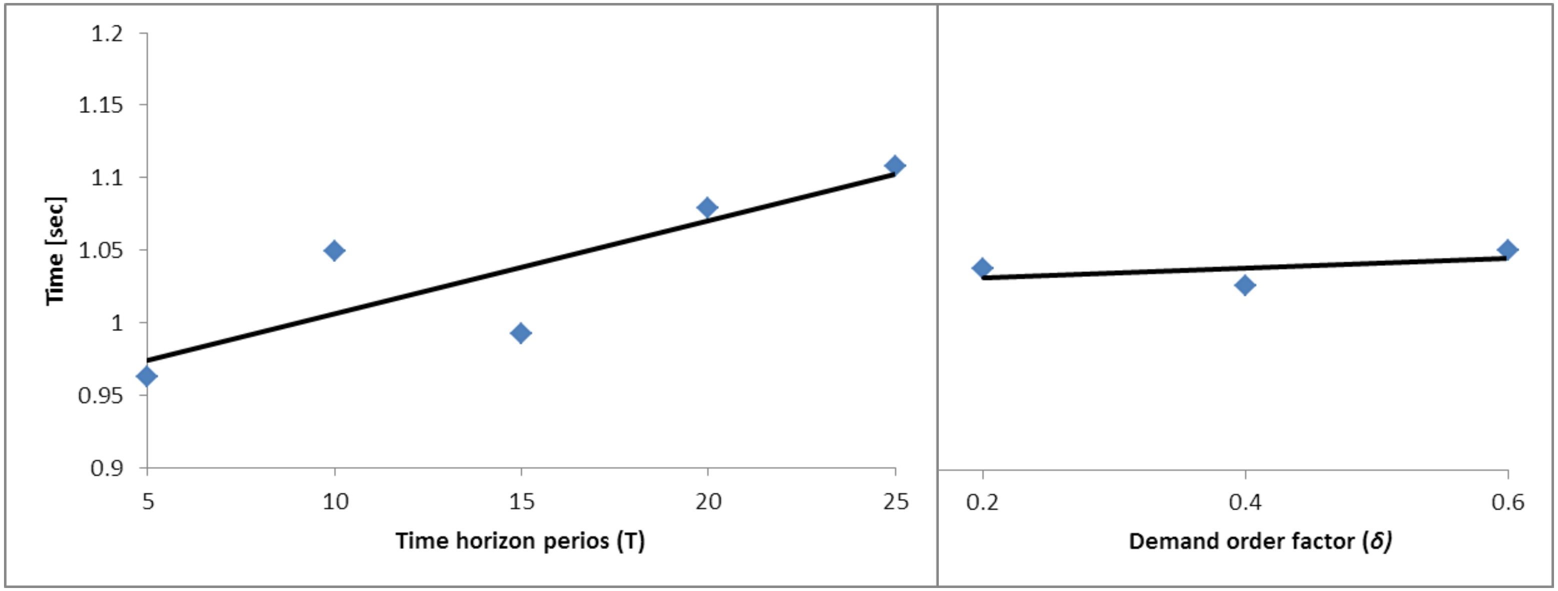
Experimental Results:
Optimal solutions are affected by T and δ



74% of solution ≤1 time period differences from Optimal.



Computational Time does not increase exponentially with the problem size



Summary of ANOVA tests

	δ	T
Proportion of Optimal Solution	5.70 (0.004)	4.02 (0.003)
Absolute Difference	5.87 (0.003)	4.66 (0.001)
CPU Time [sec]	0.30 (0.741)	4.35 (0.002)

F-statistic (P-value)

Questions

References

W. Vanderbloemen, "*Forbes*," *This Is Why People (Not Technology) Are Still Your Greatest Asset*, Dec-2016.

Shafer, S. M., Nembhard, D. A., & Uzumeri, M. V. (2001). The effects of worker learning, forgetting, and heterogeneity on assembly line productivity. *Management Science*, *47*(12), 1639-1653.

Yan, J. H., & Wang, Z. M. (2011, September). GA based algorithm for staff scheduling considering learning-forgetting effect. In *Industrial Engineering and Engineering Management (IE&EM), 2011 IEEE 18Th International Conference on* (pp. 122-126). IEEE.

Biskup, "A state-of-the-art review on scheduling with learning effects," *Eur. J. Oper. Res.*, vol. 188, no. 2, pp. 315–329, 2008.

W. Wu, "Solving a two-agent single-machine learning scheduling problem," *Int. J. Comput. Integr. Manuf.*, vol. 27, no. 1, pp. 20–35, 2014.

D. A. Nembhard and F. Bentefouet, "Parallel system scheduling with general worker learning and forgetting," *Int. J. Prod. Econ.*, vol. 139, no. 2, pp. 533–542, 2012.

M. Hewitt, A. Chacosky, S. Grasman, and B. Thomas, "Integer Programming Techniques for Solving Non-linear Programs with an Appl

H. Jin, B. W. Thomas, and M. Hewit, "Integer programming techniques for makespan minimizing workforce assignment models that recognize human learning," *Comput. Ind. Eng.*, vol. 97, pp. 202, 2016.

C. Wang, C. Liu, Z. H. Zhang, and L. Zheng, "Minimizing the total completion time for parallel machine scheduling with job splitting and learning," *Comput. Ind. Eng.*, vol. 97, pp. 170–182, 2016.

V. V. Prabhu, "Performance of real-time distributed arrival time control in heterarchical manufacturing systems.," *IIE Trans.*, vol. 32, no. 4, pp. 323–331, 2000.

Additional Slides: Empirical Data Used

Three-parameter Hyperbolic Learning model

Empirical data presented by Shafer et al. (2001) and implemented by Nembhard & Bentefouet (2012).

The mean vector and variance-covariance matrix of [*ln(k), ln(p), ln(r)*] are approximated by **μ** and **∑** as:

$$\boldsymbol{\mu} = [\,3.36 \;\; 4.66 \;\; 4.83] \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 0.054 & 0.397 & 0.374 \\ 0.397 & 7.821 & 5.430 \\ 0.374 & 5.430 & 4.923 \end{bmatrix}$$

- $\boldsymbol{p_{i,j}}$ represented the initial productivity level of worker *i* at task *j*
- $\boldsymbol{r_{i,j}}$ the learning rate parameter of worker *i* at task *j*
- $\boldsymbol{k_{i,j}}$ the steady state production rate of worker *i* at task *j.*

$$f_{i,j}(t) = k_{i,j} \frac{t + p_{i,j}}{t + p_{i,j} + r_{i,j}}$$

Additional Slides: Algorithm’s Inputs

Problem parameters:

- Production time horizon (*T*)
- Demand order size of both tasks required at the end of the time horizon ($D_j \ \forall\ j$)
- Output of worker *i* performing task *j* during period *t* ($O_{i,j,t}$).

Heuristics parameters :

- *Arrival time controller* law gain (α),
- *Switching time controller* law gain (γ),
- Upper bound for the discrete event simulation iterations (θ).

The values of these parameters can have a direct impact on the stability of the heuristic and its performance on generating good schedules , as shown by Prabhu (2000).

Based on initial experimental results these parameter were set:

$$\boldsymbol{\alpha = 0.2, \gamma = 0.4,\ \textit{and}\ \theta = 1{,}000}.$$

Additional Slides: Algorithm

Input: $T, D_j, O_{i,j,t}, \alpha, \gamma, \theta \quad \forall i \ \forall j, \forall t$;

1. **Step 1:** Define:
Simulation iteration $r = 0$, Arrival vector $a(\theta) = \emptyset$ with it first entry $a(1) = 1$, Switching vector $s(\theta) = \emptyset$ with it first entry $s(1) = 1$, Task assignment $\omega_1 = 1, \omega_2 = 2$, Tasks production output vector $P_1(\theta) = \emptyset$ and $P_2(\theta) = \emptyset$;

1. **Step 2:** Select the initial worker assignment;

if $\sqrt{\frac{\left(\sum_{t=1}^{T} x_{2,1,t} * O_{2,1,t} - D_1\right)^2 + \left(\sum_{t=1}^{T} x_{1,2,t} * O_{1,1,t} - D_2\right)^2}{2}} < \sqrt{\frac{\left(\sum_{t=1}^{T} x_{1,1,t} * O_{1,1,t} - D_1\right)^2 + \left(\sum_{t=1}^{T} x_{2,2,t} * O_{2,2,t} - D_2\right)^2}{2}}$ **then**

$\omega_1 \leftarrow 2$ and $\omega_2 \leftarrow 1$;

end

1. **Step 3:** Applied switching and arrival controllers;

2. $P_1(r) \leftarrow \sum_{t=a(r)}^{s(r)-1} x_{\omega_2,1,t} * O_{\omega_1,1,t} + \sum_{t=s(r)}^{T} x_{1,1,t} * O_{\omega_2,1,t}$,

3. $P_2(r) \leftarrow \sum_{t=a(r)}^{s(r)-1} x_{\omega_1,2,t} * O_{\omega_1,1,t} + \sum_{t=s(r)}^{T} x_{\omega_2,2,t} * O_{\omega_2,1,t}$;

4. **if** $P_1(r) \geq D_1$ & $P_2(r) \geq D_2$ **then**

5. $a(r+1) \leftarrow a(r) + \alpha\left[\min\left(\frac{P_1(r)-D_1}{(0.5)*(P_1(r)+P_2(r))}, \frac{P_2(r)-D_2}{(0.5)*(P_1(r)+P_2(r))}\right)\right]$, **else** $a(r+1) \leftarrow a(r)$;

6. **if** $P_1(r) < D_1$ or $P_2(r) < D_2$ **then**

7. $s(r+1) \leftarrow s(r) + \gamma\left[\min\left(\frac{D_1-P_1(r)}{(0.5)*(P_1(r)+P_2(r))}, \frac{D_2-P_2(r)}{(0.5)*(P_1(r)+P_2(r))}\right)\right]$, **else** $s(r+1) \leftarrow \min(s(r), a(r))$;

8. **Step 4:** Repeat Step 3 and 4 with $r \leftarrow r + 1$ **until** $r + 1 = \theta$

9. **Step 5:** Output : $\omega_1, \omega_2, a(m), \quad s(m), \ x_{i,j,t} \quad \forall i \ \forall j, \forall t$;

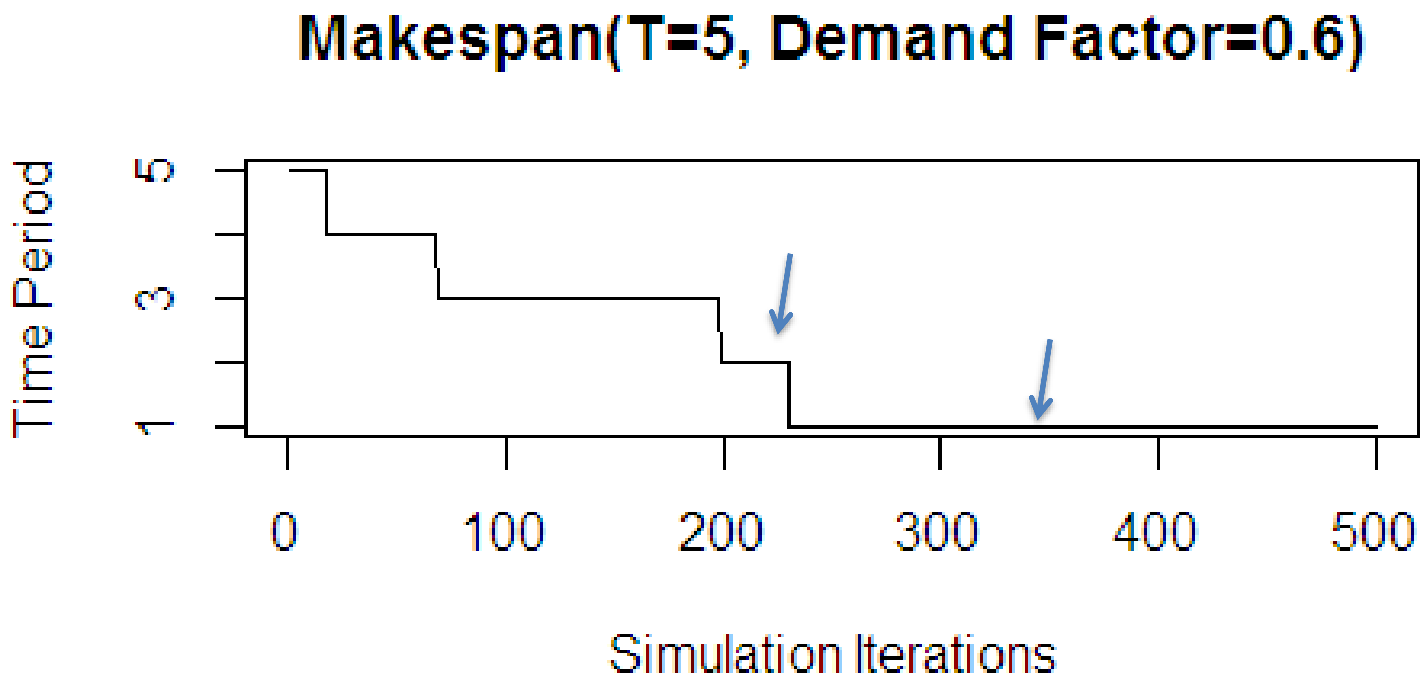
$a(m) = \min(a(\theta) | P_1(\theta) \geq D_1 \ \& \ P_2(\theta) \geq D_2)$

$x_{\omega_1,1,t}$ = 1 for $a(m) < t < s(m-1)$, $x_{\omega_1,2,t}$ = 1 for $s(m) < t < T$, 0 otherwise

$x_{\omega_2,2,t}$ = 1 for $a(m) < t < s(m-1)$, $x_{\omega_2,1,t}$ = 1 for $s(m) < t < T$, 0 otherwise

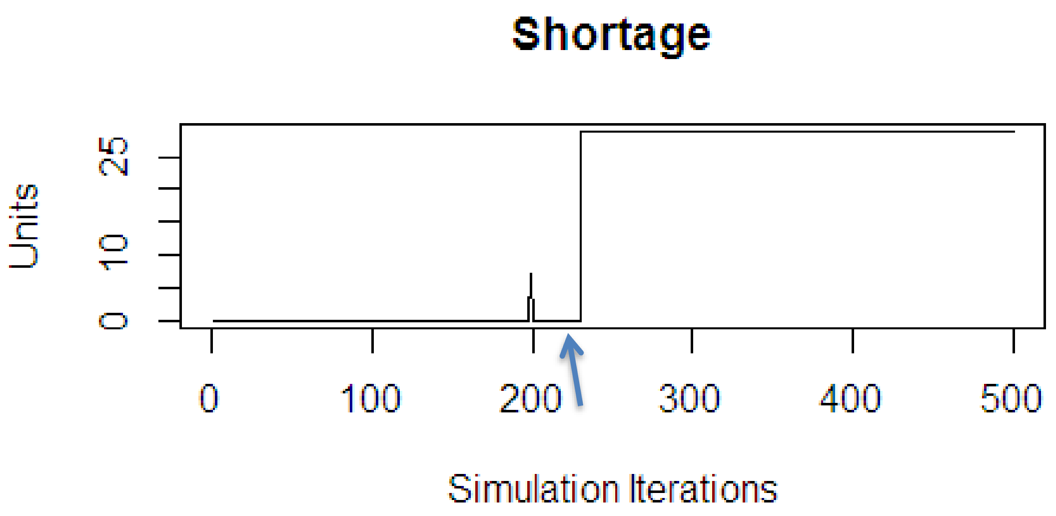
Additional Slides: Algorithm's Outputs

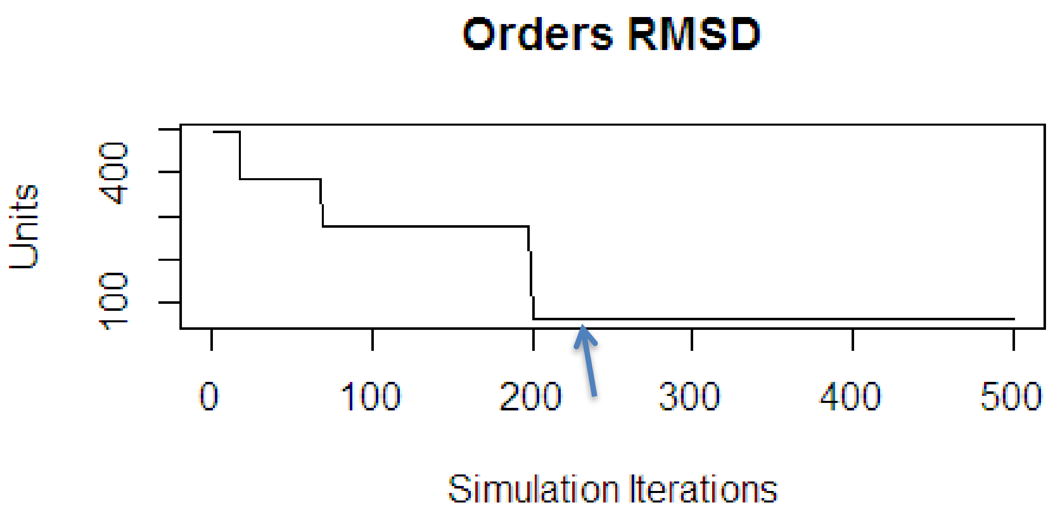
Example 1



Additional Slides: Algorithm’s Outputs

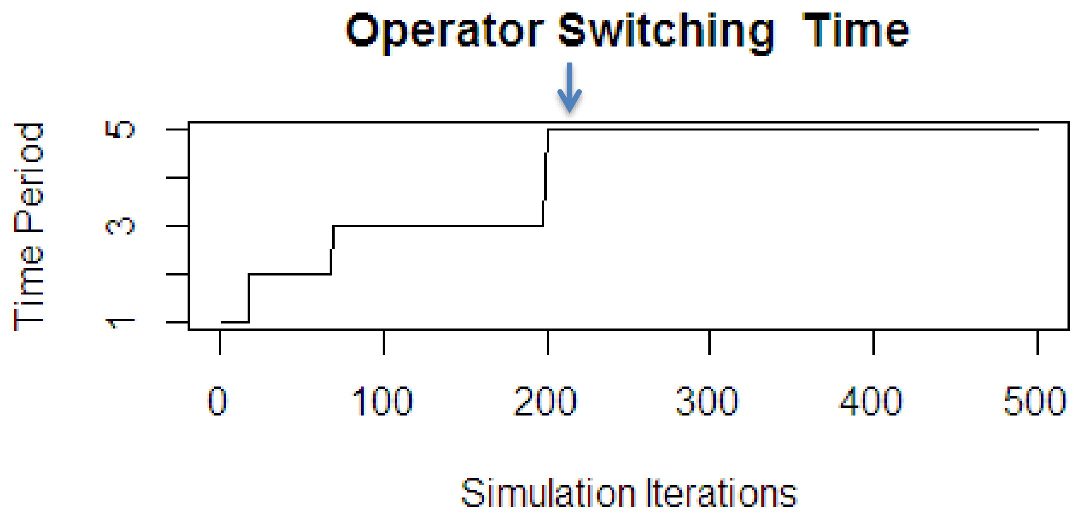
Example 1

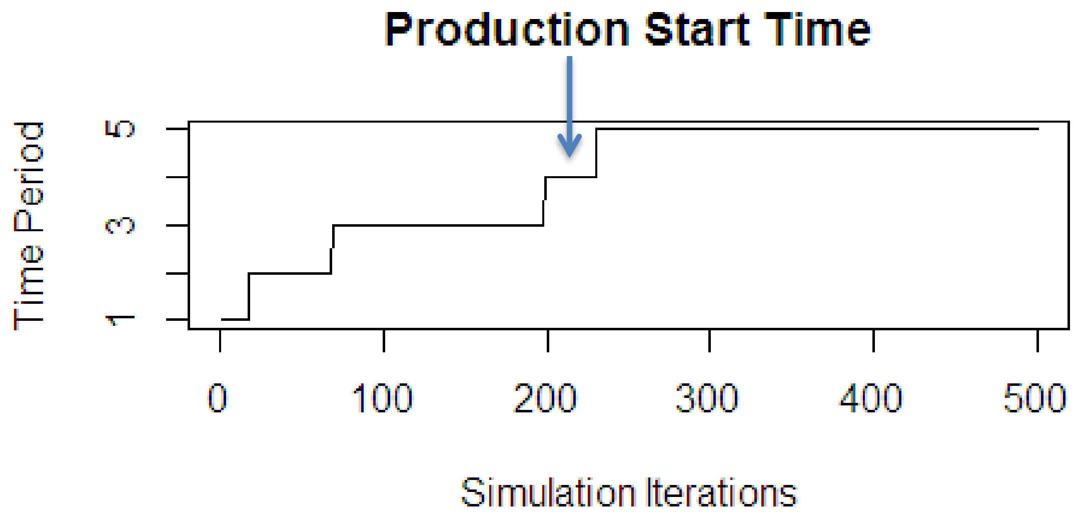




Additional Slides: Algorithm's Outputs

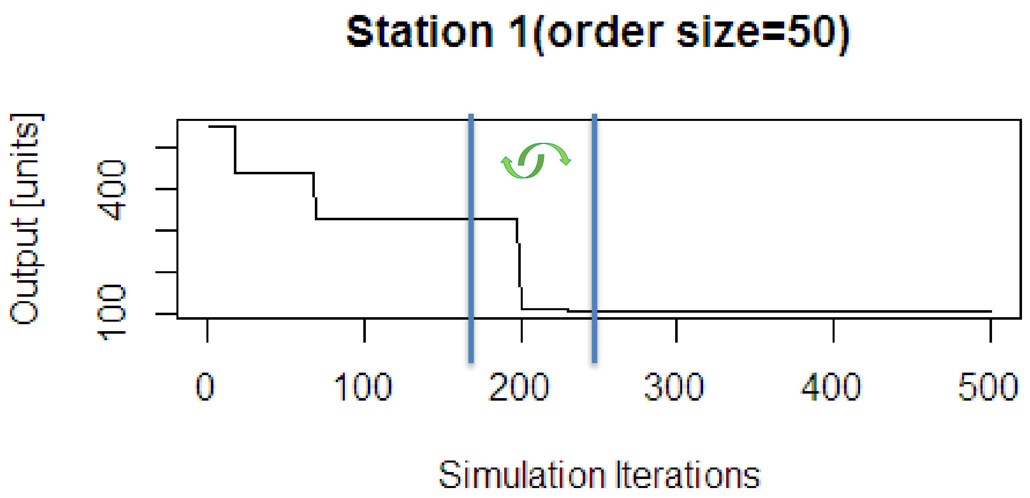
Example 1

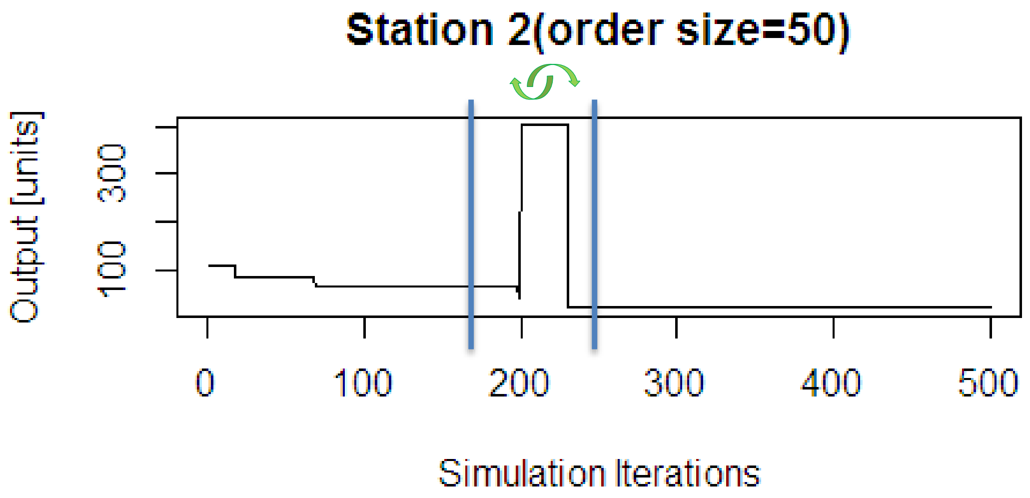




Additional Slides: Algorithm's Outputs

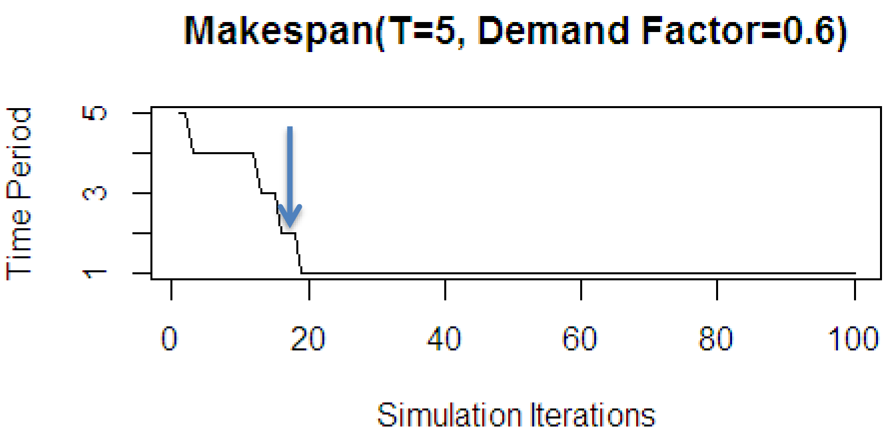
Example 1

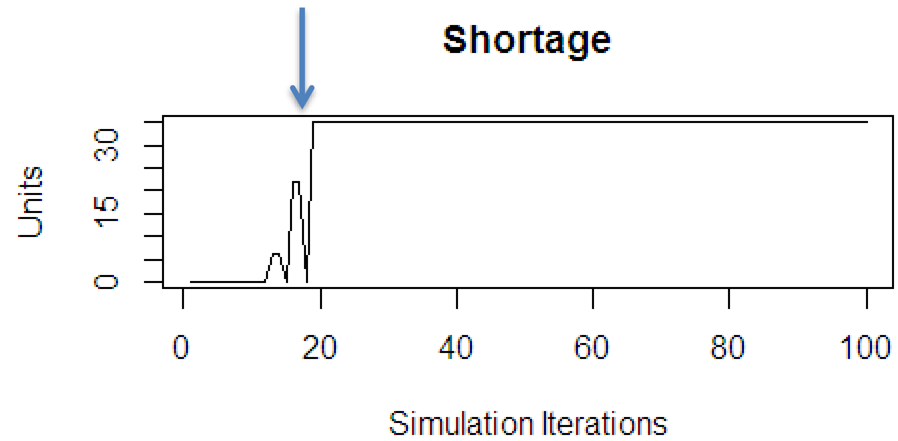


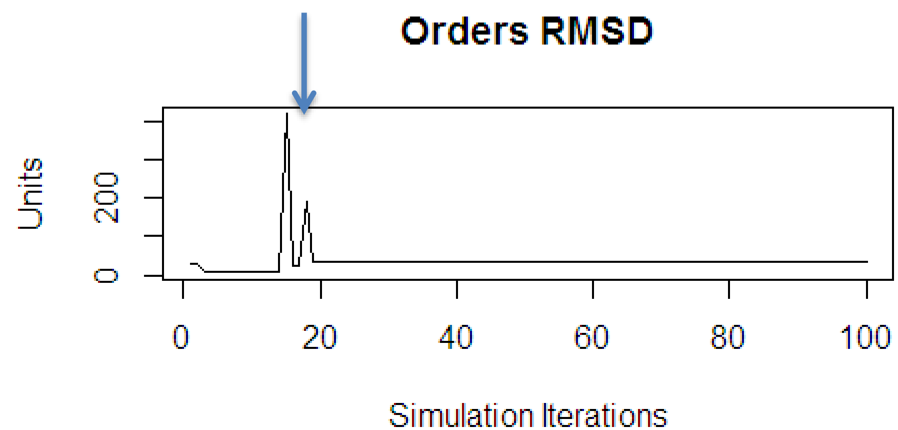


Additional Slides: Algorithm's Outputs

Example 2







Additional Slides: Algorithm's Outputs

Example 2