Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

ROLLING HORIZON SCHEDULING OF MULTI-FACTORY SUPPLY CHAINS

Eunkyoung G. Cho
Kristin A. Thoney

College of Textiles
North Carolina State University
Box 8301
Raleigh, NC 27695, U.S.A.

Thom J. Hodgson
Russell E. King

College of Engineering
North Carolina State University
Box 7906
Raleigh, NC 27695, U.S.A.

ABSTRACT

The Virtual Factory is a job shop scheduling tool that was developed at NC State. It has been found to provide near-optimal solutions to industrial-sized problems in seconds. Recently, the Virtual Factory was expanded to include inter-factory transportation operations which enabled the detailed scheduling of entire multi-factory manufacturing supply chains. Separately, a rolling horizon procedure was developed to test the Virtual Factory for single factory problems. This procedure allowed us to more accurately predict how the Virtual Factory would perform in industry. Consequently, the rolling horizon procedure was extended to multi-factory settings to gauge industrial performance and eliminate transient effects found in previous multi-factory experimentation. Experimental results, under a variety of different scenarios, indicate that the Virtual Factory also performs well in multi-factory, rolling horizon settings.

1 INTRODUCTION

Recently, opportunities for cycle time reduction which exist on an inter-organizational basis have received much attention. The potential for improvement appears to be even greater in an inter-organizational supply chain environment compared to an intra-organizational environment. Most of this analysis has been done at the macro level. Little research exists on the impact of coordinated, detailed production scheduling between different entities in the supply chain and the intermediate transportation.

Hodgson et al. (1998, 2000) developed a job shop scheduling algorithm and named it the Virtual Factory (VF). The VF is an iterative, simulation-based procedure, whose objective is minimizing maximum lateness. It has been found to provide near-optimal solutions to industrial-sized problems in seconds.

Thoney et al. (2002a) expanded the VF to include inter-factory transportation operations which enabled the detailed scheduling of entire multi-factory manufacturing supply chains. Although performance was found to be good when transportation was not a bottleneck, the scenarios were tested in a transient setting. Starting and ending effects were observed to impact performance.

The more realistic rolling horizon setting explained in Thoney et al. (2002b) would enable us to more accurately test how the VF would perform in multi-factory settings in industry by helping to eliminate transient effects. Using the rolling horizon algorithm, a variety of experiments were undertaken to gauge performance under different conditions.

In Section 2, the original VF is introduced. A discussion of incorporating batch processors into the VF and a description of the rolling horizon procedure are found in Sections 3 and 4, respectively. Section 5 explains the problem generation. In Sections 6 and 7, experimental results for *Two Factories in Series*, *Three Factories in Series*, and *Two Factories Feeding One* are discussed. Section 8 presents the conclusions and future research.

2 VIRTUAL FACTORY

The Virtual Factory consists both of a scheduling algorithm and a lower bound.

2.1 Scheduling Procedure

Let d_i be the due date of job i and p_{ij} be the processing time of job i on machine j. Then the slack of job i on machine m is calculated as

$$Slack_{i,m} = d_i - \sum_{j \in m+} p_{ij} \quad (1)$$

where $m+$ is the set of all operations subsequent to machine m on job i's routing. Slack represents the latest possible time that a job can finish on a machine and still satisfy its final due date. As this does not include queuing time, slack did not perform well as a dispatching rule in early experiments found in the scheduling literature.

To remedy this situation, a revised slack value that incorporates queuing times is used as the sequencing rule in the Virtual Factory. Queuing times are recorded for each job at each machine it visits in one iteration of the simulation and used in the next iteration. The revised slack for job i on machine m is computed as

$$Slack'_{i,m} = d_i - \sum_{j \in m+} p_{ij} - \sum_{j \in m++} q_{ij} \qquad (2)$$

where m++ is the set of all subsequent operations to machine m on the routing sheet for job i, except the immediate subsequent operation. The simulation is run until the lower bound is achieved or a specified number of iterations is reached, and the best solution is saved.

2.2 Lower Bound

Hodgson et al. (1998, 2000) chose to evaluate the quality of the schedules produced by the VF through comparison to a lower bound (*LB*). The lower bound is calculated by decomposing the job shop problem into individual one machine problems. To do this, an earliest start time and a latest finish time are calculated for each machine on each job's route. Let r_i be the release time of job i. Then the earliest possible start time for a job i on machine m is,

$$ES_{i,m} = r_i + \sum_{j \in m-} p_{ij} \qquad (3)$$

where $m-$ is the set of all operations preceding machine m on job i's routing sheet. The latest finish time for each job i on machine m is

$$LF_{i,m} = d_i - \sum_{j \in m+} p_{ij} \qquad (4)$$

where $m+$ is the set of all operations following machine m on the routing sheet of job i.

The lower bound for the job shop problem ($N/M/L_{max}$) is obtained by solving the $N/1/\ L_{max} \mid r_i$ problem on each machine m by considering $LF_{i,m}$ as the effective due date for job i on machine m and $ES_{i,m}$ as the release time (r_i) for job i on machine m. Since $N/1/\ L_{max} \mid r_i$ is NP-hard, a relaxation suggested by Baker and Su (1974) is used. The relaxation is to allow preemption of a job in process whenever one with a more imminent due date becomes available.

The overall lower bound, $LB\ (L_{max})$, is computed as

$$LB(L_{\max}) = \max_{m=1,M} \{LB_m(L_{\max})\} \qquad (5)$$

where $LB_m\ (L_{max})$ is the lower bound for machine m. The power of this lower bound is that there are M chances to get a tight bound.

3 BATCH TRANSPORTATION PROCESSORS

Thoney et al. (2002a) generalized the sequencing procedure and *LB* to include batch processors within the VF.

3.1 Queuing Schemes

The manner in which queuing time is used in the VF is inappropriate for batch operations because prioritizing the jobs by increasing revised slack does not account for the way they interact in batch processing. Two methods of incorporating batch processors into the procedure were discussed. The following notation is required.

- Q_i - Queuing time for job i at the batch processor;
- JA_i - Time job i arrives at batch processor's queue;
- BF_i - Time the machine on which job i is processed finished its previous batch;
- BA_i - Time the next batch begins processing after the arrival of job i;
- BB_i - Time the previous batch begins processing be-fore the arrival of job i.

It is important to note that BB_i and BF_i may or may not refer to the time at which the same batch processor begins and ends processing. The queuing schemes are as follows.

Queuing Scheme 1. $Q_i = \max\{JA_i - BF_i,\ 0\}$

Queuing Scheme 2. $Q_i = \begin{cases} JA_i - BB_i & \text{if } JA_i \le (BA_i - BB_i)/2 \\ 0 & \text{otherwise} \end{cases}$

Queuing Scheme 1 tries to force the machine to begin processing as soon as possible, and Scheme 2 tries to place a job in the right batch by giving a higher priority in the next iteration to jobs that arrive before the midpoint of processing of the previous and next batch. Since Scheme 2 tended to dominate Scheme 1, Scheme 2 was used for subsequent experiments.

3.2 Lower Bound

To be able to use the lower bound for the VF, a lower bound on transportation was devised. Transportation is a problem of batch processors in parallel. This problem is known to be NP-hard. Therefore, the problem was relaxed for efficient computation. A capacity relaxation was first performed, transforming the problem of batch processors in parallel to parallel machines. Then, preemption was allowed, as well as an additional processing relaxation on the resulting preemptive parallel machine problem. Finally, the lower bound was improved in several ways. These enhancements were based on the characteristics of solutions to the original batching problem that had been lost in the relaxation process leading to the lower bound (i.e., release times, due-dates, and capacity profiles). See Thoney et al. (2002a) for more details.

It is important to note that this lower bound is valid regardless of what operational policy is used to begin processing. In other words, it is valid if processing only begins with full batches or if processing can begin with partial batches. In the multi-factory VF experiments, trucks do not leave a factory until they are full. This policy was implemented since it minimizes transportation costs.

4 ROLLING HORIZON SETTING

Most sequencing algorithms are evaluated in a transient setting in the scheduling literature. This does not adequately reflect how scheduling systems are used in industry. Plants usually contain many different orders, with new orders arriving as older ones are completed. Scheduling is often performed on some regular basis. The best schedule is implemented until the plant is rescheduled. Thus scheduling occurs on a rolling horizon basis. Consequently, Thoney et al. (2002b) developed a rolling horizon scheduling procedure to more accurately test the VF for single factory problems.

The following definitions are required for this section:

- t - Current time in days
- c_j - Completion time of job j
- N - Total number of jobs
- N_s - Total number of jobs starting in factory on first day
- M - Total number of machines
- UL - Upper limit of uniform distribution for number of operations
- JR - Number of jobs released each day
- RO - Number of operations for jobs released
- DL - Length of a day
- T - Total horizon length in days
- w - Number of days in warm-up period
- WIP - Work in process (number of days)
- i - Number of iterations
- $\overline{Mops}$ - The average number of operations that a single machine can process in a day
- $\overline{Ops}$ - The average number of operations that each job which starts in the factory has
- $\overline{P}$ - The expected operation processing time.

4.1 Scheduling Procedure

The algorithm for the rolling horizon scheduling procedure is given as follows:

```
1. Initialize t = 0
    1.1 If t = w + 1, compute LB
    1.2 Release jobs whose rj = t
    1.3 Run the Virtual Factory i iterations
    1.4 Implement the first day of the best schedule
    1.5 t = t + 1
    1.6 Continue from 1.1 until t = T
2. Run the remainder of the best schedule until all jobs are finished
3. Initialize j =1
    3.1 If cj > w, determine if job j is the Lmax job
    3.2 j = j + 1
    3.3 Continue from 3.1 until j = N
```

Step 1 releases the new jobs into the system, runs the VF, and implements the first day of the best schedule. This procedure is repeated each day until the total number of days is reached. In step 2, the best schedule is run until all jobs are finished. This ensures that the scheduling procedure did not sacrifice the remaining jobs in the factory to yield a good schedule. In Step 3, the lateness for each job completed after the warm-up period is compared to the current maximum lateness.

4.2 Lower Bound

The *LB* is computed in the same manner as for the original VF, except that the *LB* for the rolling horizon schedule is computed after the warm-up period. The *LB* calculation includes both jobs that are currently in the factory after the warm-up period, with their remaining operations and processing times, and also those jobs that are released later during the complete horizon of the simulation. Therefore, even though there are multiple runs of the VF engine for the rolling horizon scheduling procedure, there is only one *LB* calculation.

4.3 Calculation of JR and Ns

To balance input orders and output products, the number of jobs released each day and the total number of jobs starting in the factory on the first day needs to be calculated. A well-defined calculation procedure was developed by Thoney et al. (2002b). *JR* is calculated by following formula:

$$JR \approx \frac{(M)(\overline{Mops})}{RO}, \tag{6}$$

$$\text{where } \overline{Mops} = \frac{DL}{\overline{P}}. \tag{7}$$

N_s is computed by following formula:

$$N_s \approx \frac{(JR)(RO)(WIP)}{\overline{Ops}}, \tag{8}$$

$$\text{where } \overline{Ops} = (UL+1)/2. \tag{9}$$

5 PROBLEM GENERATION

Scenarios of *Two Factories in Series*, *Three Factories in Series*, and *Two Factories Feeding One* are considered. These scenarios are based on the original models in Thoney et al. (2002a). The jobs that start in the factories are generated as described in the following sections, and the jobs that are released into the system after time 0 are generated as jobs with the maximum number of operations.

5.1 Two Factories in Series

The scenario is depicted in Figure 1.

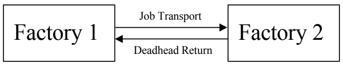

Figure 1: Two Factories in Series (Thoney et al. 2002a)

Jobs are each assigned a number of operations, *n,* which is randomly generated, and a corresponding (Uniformly distributed) due-date. The number of operations is distributed Uniform [1,7]. If $n > 4$, then the job is processed on $n - 4$ machines in Factory 1, transported by truck to Factory 2, and processed on three machines in Factory 2. If $n = 4$, then the job is transported by truck to Factory 2, and processed on three machines in Factory 2. If $n < 4$, then the job is processed on *n* machines in Factory 2.

5.2 Three Factories in Series

The situation is illustrated in Figure 2.

Figure 2: Three Factories in Series (Thoney et al. 2002a)

Jobs are each assigned a number of operations, *n,* that is distributed Uniform [1,11] and a corresponding due-date. If $n > 8$, then the job is processed on n -8 machines in Factory 1 and processed on three machines in Factory 2 and Factory 3. If $n = 8$, then the job is transported by truck to Factory 2, and processed on three machines in Factory 2 and Factory 3. If $4 < n < 8$, then the job is processed on n - 4 machines in Factory 2 and processed on three machines in Factory 3. If $n = 4$, then the job is transported by truck to Factory 3, and processed on three machines in Factory 3. If $n < 4$, then the job is processed on *n* machines in Factory 3.

5.3 Two Factories Feeding One

The scenario is depicted in Figure 3.

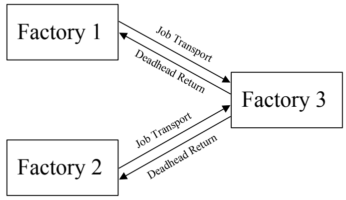

Figure 3: Two Factories Feeding One (Thoney et al. 2002a)

It represents an assembly operation where a specific job from Factory 1 and a specific job from Factory 2 are assembled into a job that is processed in Factory 3. A random number of operations, *n*, that is distributed Uniform[1,7] is generated. If $n > 4$, then 3 jobs are generated. The first job is processed on $n - 4$ machines in Factory 2 and transported by truck to Factory 3. The second job is processed on $n - 4$ machines in Factory 2 and transported by truck to Factory 3. The third jobs represents the assembly of 1 and 2 and is processed on 3 machines in Factory 3. If $n = 4$, 3 jobs are also generated. The only difference is that the first and second job are not processed in Factory 1 and 2, respectively. If $n < 4$, then 1 job is generated. It is processed on *n* machines in Factory 3.

5.4 Simulation

The warm-up period in days, *w*, was set equal to 10 since this is significantly larger than the *WIP* in the problem. Each problem was tested over a variety of due date ranges since it is a factor known to influence solution performance. A due date range, *DDR*, is defined so that each job, *j*, is randomly generated a discrete uniform due date between r_j and $r_j + DDR$, where $r_j=0$ for jobs initially in the factory. Experiments were run for due date ranges between 0 and 25 days. For each due date range, 10 replications were run and the average difference between L_{max} and *LB* was calculated. This difference is the maximum by which the simulation solution could exceed the optimal solution. A positive difference between L_{max} and *LB* could be the result of a non-optimal schedule, a weak *LB*, or a combination of the both.

6 TWO FACTORIES IN SERIES

6.1 Base Case

In this paper, problems with 25 machines in each factory were considered. For job route generation details, refer to Section 5.1. In each problem, *DL* =1600 and *i*=100. *RO* and *WIP* were set equal to 3. One way travel time for trucks was 200. Truck capacity was 25 jobs and the total horizon length was set at 100 days.

The number of jobs released each day and total number of jobs starting in the factory on the first day were calculated. Since the processing times for the problems are uniformly distributed between 1 and 200, $\overline{P} \approx 100$ and thus $JR \approx (25)(16)/3=133.33$. This value tends to overestimate JR since it assumes that there is never any idle time on the machines. Therefore, experimentation was performed to determine the actual value of JR, starting with the computed value. JR was found to be 120 and $N_s \approx [(133.33)(3)(3)]/2 \approx 600$.

6.2 Experimentation of Queuing Schemes

The queuing schemes were tested on the *Two Factories in Series* base case. The performance of Scheme 1, $L_{max} - LB$, is compared to that of Scheme 2. Unlike the results in the transient setting (Thoney et al., 2002a), there is no significant dominance between Queuing Scheme 1 and Queuing Scheme 2 (Figure 4.).

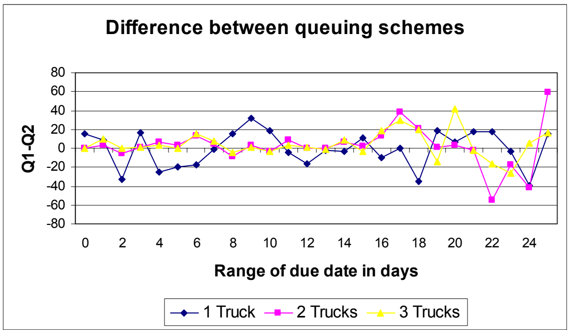

Figure 4: Comparison of Queuing Schemes

Since Scheme 2 tended to dominate Scheme 1 in Thoney's research, Scheme 2 was used for her subsequent experiments. In this study, Scheme 2 will be used for all subsequent experiments.

6.3 Performance Observations of Base Case

In Figure 5, the difference between the attained L_{max} and lower bound (LB) is displayed as a function of the due date range and number of trucks.

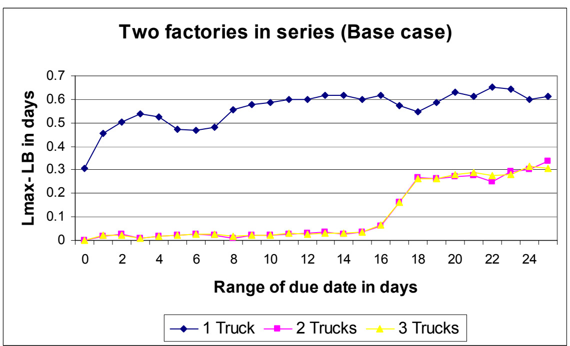

Figure 5: Performance of the Base Case of *Two Factories in Series*

For greater than two trucks, transportation was not a bottleneck. For 1 truck, performance is good, but it is better for 2 and 3 trucks, particularly in the lower due date ranges. To put these differences in perspective, recall that 90 days of factory performance were included in these statistics, with the latenesses of over (90)(120) = 10,800 jobs taken into account.

As in the transient experiments, when trucks are no longer a bottleneck, performance is very similar no matter how many trucks are added. The calculations for estimating this point in rolling horizon scenarios are closely related to those used for transient problems [Thoney et al, 2002a] after calculating the rate at which jobs leave Factory 1. Since the factory input was balanced with the factory output when 120 jobs were released each day, a job finishes on average every $1600/120 \cong 13.33$ units of time. Therefore, on average, a new truckload (capacity = 25) is ready for loading in Factory 1 every 13.33(25)=333.25 time units. To handle this rate, when the round trip travel time for each truck is 400 time units, approximately $\lceil 400 / 333.25 \rceil = 2$ trucks are needed. This is consistent with the observed data.

Recall that each data point on our graphs represents the average of 10 problems. Each of the problems in Figure 5 where transportation was not a bottleneck took an average of 158.83 seconds to run on a 2 GHz Pentium. Of this time, approximately 11.1 seconds was used in computing the lower bound, and it took about 1.48 seconds to schedule and implement the best sequence for each of the 100 days in the horizon. For problems where transportation was a bottleneck, the computation time was longer due to the build up of jobs in the system. The computational effort required for the remaining experimentation in this paper is analogous, with the computation time per day increasing approximately linearly with the problem size.

6.4 Varying Truck Capacity

Truck capacity was varied to be 10 and 20 jobs in situations where transportation is and is not a bottleneck to determine how scheduling performance is affected. In Figures 6 and 7, the difference between the attained L_{max} and lower bound (LB) with truck capacity of 10, capacity of 20 and the base case are displayed as a function of the due date range and number of trucks. In Figure 6, the results of varying capacity when the truck is a bottleneck are displayed. Two trucks of capacity 10 performed the same as one truck of capacity 20. One truck of capacity 20 performed better than one truck of capacity 25.

One explanation for the difference in performance between 1 truck of capacity 20 and 1 truck of capacity 25 is that there is extra waiting time with 1 truck of capacity 25, since it was assumed that a truck does not leave until it is full. This extra waiting time is not taken into account in the *LB*.

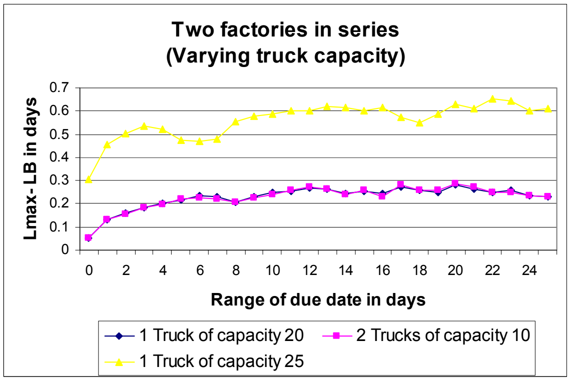

Figure 6: Performance with Varying Truck Capacity in *Two Factories in Series*

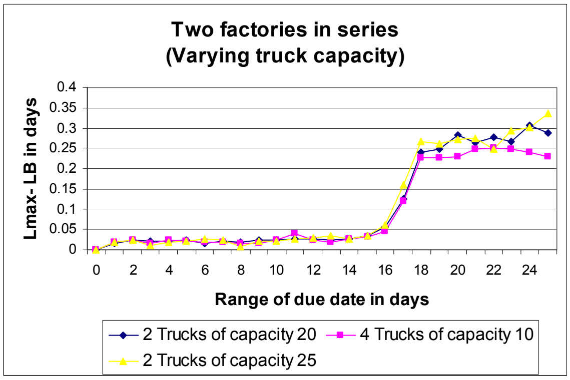

Figure 7: Performance with Varying Truck Capacity in *Two Factories in Series*

Figure 7 shows the results of varying capacity when trucks are not a bottleneck. Four trucks of capacity 10 performed slightly better than two trucks of capacity 20 in the high due date range. Two trucks of capacity 20 performed similar to two trucks of capacity 25. The extra time incurred waiting for trucks of capacity 25 to leave versus trucks of capacity 20 is not as significant in lateness calculations when the trucks are not always the bottleneck.

6.5 Varying the Total Horizon Length

To determine the effect of the total number of days that are scheduled on the quality of the scheduling solutions, each problem was run for 55 days and 190 days with the same 10 day warm-up. This will allow us to compare performances for half and twice as long as the base case. In Figure 8, the difference between the attained L_{max} and lower bound (LB) with varying the total horizon length, T, is displayed as a function of the due date range and number of trucks. Trucks have a capacity of 25 jobs and the number of trucks is three in each case.

The average differences between L_{max} and *LB* of all the three settings were the same in the low due date ranges. Performance got slightly worse as the horizon length was

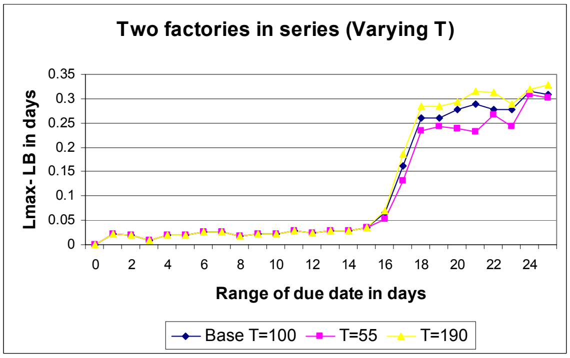

Figure 8: Performance with Varying the Total Horizon Length in *Two Factories in Series*

increased in the high due-date ranges (i.e., after 16 days). The performance gap is about 15 minutes in the low due date ranges, assuming one 8-hour shift per day.

7 OTHER MULTI-FACTORY PROBLEMS

The performance of *Three Factories in Series* and *Two Factories Feeding One,* while not completely identical to *Two Factories in Series,* is similar. Thus graphical results are not presented here except for the base case of each scenario. Recall that the number of trucks displayed on each graph refers to the number of trucks that are transporting jobs between each location.

7.1 Three Factories in Series

In this problem, the parameters are the same as in the base case in *Two Factories in Series*, with the exception that the maximum number of operations is 11. In Figure 9, the difference between the attained L_{max} and lower bound (LB) is displayed as a function of the due date range and number of trucks. *Three Factories in Series* performs very well when transportation is not a bottleneck. Compared to *Two Factories in Series*, it performs even better in the high due date ranges.

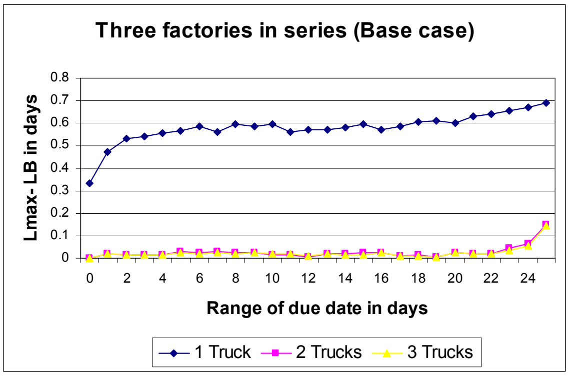

Figure 9: Performance of Base Case of *Three Factories in Series*

7.2 Two Factories Feeding One

In this problem, all parameters are the same as the base case in *Two Factories in Series*. In Figure 10, the difference between the attained L_{max} and lower bound (LB) is displayed as a function of the due date range and number of trucks.

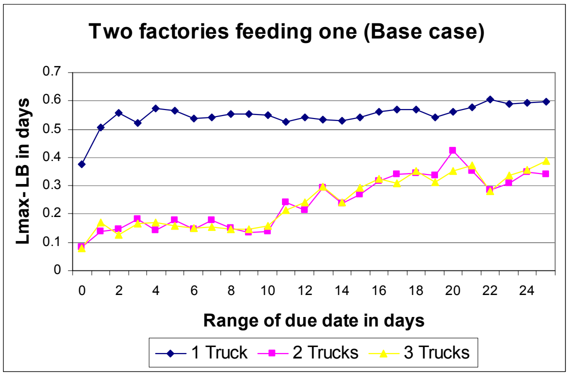

Figure 10: Performance of Base Case of *Two Factories Feeding One*

For greater than two trucks, transportation was not a bottleneck. Performance was close to the lower bound in the low and middle due-date ranges and gradually became slightly worse (i.e., after 10 days). In the problem generation, it was assumed that a certain job from Factory 1 had a matched job in Factory 2. Once both of the matched jobs arrived at Factory 3, they were replaced by a single job assembly. This potential added waiting time was not taken into account in the lower bound.

8 CONCLUSIONS AND FUTURE RESEARCH

The Virtual Factory has been shown to perform well in multi-factory, rolling horizon settings. *Two Factories in Series*, *Three Factories in Series*, and *Two Factories Feeding One* were discussed. In all cases, the difference between the maximum lateness and lower bound is relatively small with respect to the total horizon length and number of jobs processed. Performance is particularly good in cases where transportation is not a bottleneck. In the industrial settings we have observed, transportation is seldom a bottleneck and is, at worst, a transient situation.

When transportation is not a bottleneck, performance became somewhat worse in the high due-date ranges. Since the increasing difference could be the result of a *LB* that is becoming weaker, scheduling that is not as good, or a combination of the two, both the *LB* and scheduling procedure will be investigated in these situations. It is also important to run the *Three Factories in Series* experiments for even higher due date ranges because there is evidence that performance may start to worsen in this case also. In addition, experiments where transportation is a bottleneck need to be generated where the input into the system is balanced with the bottleneck to alleviate jobs building up in the system. Furthermore, experiments need to be performed verifying that transient effects have been eliminated from our performance measures. Experiments need to be conducted in which the warm-up period is longer as well as when there is a definite stopping time to eliminate possible ending effects.

REFERENCES

Hodgson, T.J., D. Cormier, A.J. Weintraub, and A. Zozom. 1998. Satisfying due-dates in large job shop. *Management Science* 44 (10): 1442-1446.

Hodgson, T.J., R.E. King , K.A. Thoney, N. Stanislaw, A.J. Weintraub, and A. Zozom. 2000. On satisfying due-dates in large job shop: idle time insertion. *IIE Transactions* 32: 177-180.

Thoney, K.A., T.J. Hodgson, R.E. King, M.R. Taner, and A.D. Wilson. 2002a. Satisfying due-dates in large multi-factory supply chains. *IIE Transactions* 34: 803-811.

Thoney, K.A., J.A. Joines, P. Manninagarajan, and T.J. Hodgson. 2002b. Rolling Horizon scheduling in large job shops. *Proceedings of the 2002 Winter Simulation Conference*, ed. E Yucesan, C.-H. Chen, J.C. Snowdon, and J.M. Charnes. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

EUNKYOUNG GLORIA CHO is a graduate student in the Textile and Apparel, Technology and Management Dept. at NC State. She received her MS and BS in Fiber and Polymer Science from Seoul National University, South Korea. You can reach her by e-mail at <echo@unity.ncsu.edu>.

KRISTIN A. THONEY, an Assistant Professor in the Textile and Apparel, Technology and Management Dept. at NC State, joined the faculty in 2000. She earned her Ph.D. in Industrial Engineering and Operations Research and her MS in Operations Research also from NC State. Kristin's research interests include production scheduling, logistics, and supply chain modeling. She is a member of INFORMS and IIE. Her email and web addresses are <Kristin_thoney@ncsu.edu> and <http://www.tx.ncsu.edu/faculty_center/directory/>.

THOM J. HODGSON, the James T. Ryan Professor of Industrial Engineering and Director of the Integrated Manufacturing Systems Engineering Institute at NC State, joined the faulty in 1983 from 13 years on Univ. of Florida's Faculty and at Ford Motor Co. He earned a BSE in science engineering, a MBA in quantitative methods and a Ph.D. in industrial engineering all from University of Michigan. Thom's research interests include production scheduling, inventory control, logistics, real-time control of systems and applied operations research. He is a member of INFORMS and IIE. His email and web addresses are <hodgson@eos.ncsu.edu> and <http://www.ie.ncsu.edu/hodgson/ >.

RUSSELL E. KING is a Professor of Industrial Engineering at North Carolina State University and has been on the faculty since 1985. Previously, he worked as a Systems Analyst for Dynamac Corporation of Rockville, Maryland and the Naval Aviation Depot at the Jacksonville Naval Air Station. He received a BSSE, Masters, and Ph.D. in IE from the University of Florida. His research interests include control and scheduling of production systems and demand activated manufacturing/supply chains. He is an associate editor of the *IIE Transactions* Focus Issue on Design and Manufacturing and is a senior member of IIE. His email and web addresses <king@eos.ncsu.edu> and <http://www.ie.ncsu.edu/king/ >.