Job Scheduling in Lean Document Production

Rong Zhou[1], **Sudhendu Rai**[2], **Minh Do**[1]

[1]Palo Alto Research Center
Palo Alto, CA 94304, USA
{rong.zhou,minh.do}@parc.com

[2]Xerox Research Center Webster
Webster, New York 14580
sudhendu.rai@xerox.com

Abstract

Lean Document Production (LDP) is a novel class of productivity-enhancement offerings that were originally invented at the Xerox Research Center Webster (XRCW) for the $100 billion printing industry in the United States. Implemented by Xerox in over 100 sites to date, LDP has provided dramatic productivity and cost improvements for both print shops and document-manufacturing facilities, as measured by reductions of 20∼40% in revenue-per-unit labor cost. LDP has generated over $200 million of incremental profit across the Xerox customer value chain since its initial introduction in 2000. In the past three years, PARC's Embedded Reasoning Area has been collaborating with XRCW to extend the scheduling capabilities of the LDP toolkit. We describe a number of newly added features such as adaptive batch splitting, multi-site scheduling and multi-core parallelization that have significantly improved the performance of our AI search-based scheduler for print shops of all sizes, particularly for those large document-production facilities that can process thousands of monthly jobs on a diverse set of document-production equipment with non-uniform speed and sequence-dependent setup times.

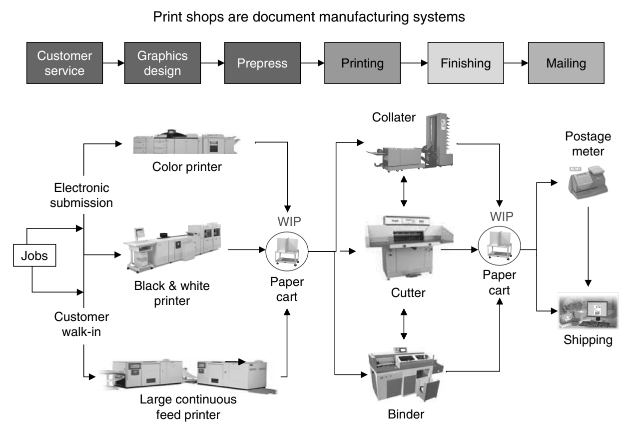

Figure 1: Print-shop workflow (WIP stands for work-in-progress).

Introduction

The provision of services that improve business productivity is a major component of Xerox's growth strategy. These services include the outsourcing and improvement of customer print-shop operations. Since 1999, Xerox has invented, tested, and implemented a novel class of productivity-improvement offerings, trademarked LDP Lean Document Production® solutions (Rai *et al.* 2009), for the printing industry. The size of the market for these offerings, which have created dramatic productivity and cost improvements for both print shops and document-manufacturing facilities, is $100 billion in the United States alone. They have greatly expanded the applications of automated scheduling tools and operations-research techniques in the printing industry.

Xerox Corporation participates in the printing industry in a number of ways; one is as a provider of services, via Xerox Managed Services (XMS), to manage print operations for clients who choose to outsource their in-plant print operations, called *in-plants*. In the 1990s, Xerox was extremely successful in growing revenues and profits in this segment by utilizing highly automated printing and reprographics equipment as a vehicle to offer print-shop outsourcing at a much lower cost than was characteristic of the typical in-plant of the day. However, other firms have since offered comparable, highly automated equipment and outsourcing services based on the use of such equipment. Thus, XMS revenues and margins came under significant pressure. LDP was conceived and invented as a set of offerings that would reestablish Xerox's reputation as the leading print-shop productivity enhancer in the industry and, in the process, increase XMS revenues and profits.

Workflow in Print-Shop Environment: Print shops can be classified into three categories based on the activity that they perform: transaction printing, on-demand publishing, or a combination of both. LDP solutions encompass all three print-shop domains. Typically, each of the six steps in the print production workflow is associated with a specific department: (1) customer service and production planning, (2) graphics design, (3) prepress, (4) printing, (5) finishing department, and (6) mailing. Any design and operations methodology for print production must comprehend both digital and off-set printing workflows independently and when they coexist. Figure 1 shows the various operations that are performed in typical print-shop workflows.

Challenges in Print-Shop Productivity Improvement

Document production has unique characteristics that make it difficult to operate print shops efficiently.

- *Long bid times*: Customers often want to see physical proofs before committing to the entire print job.
- *Variability in workflow types*: The workflow required to process print jobs varies considerably from job to job.
- *Variability in job-size distribution*: Print shops experience significant fluctuations in demand and jobs submitted to even the same shop can have sizes (e.g., number of pages) that differ by several orders of magnitude.
- *Variability in equipment capability and speed*: Print shops can host a large number of non-uniform devices, each of which may have some unique capabilities and run at various speeds under different operation modes or conditions.
- *Variability in setup times*: The setup time of even a single document-production device can change dramatically, depending on the kind of job that was processed previously on the device.
- *Variability in labor and equipment*: Print shops are often labor intensive, with many manual processing steps.
- *Departmental production and scheduling*: Print shops typically organize their equipment and labor into specific functional departments to improve utilization of resources and maintain a labor force skilled in specific tasks.

These characteristics pose significant challenges for developing a standardized productivity-improvement methodology that is scalable and adaptable across multiple printing environments.

LDP Scheduling

In this section, we describe how LDP selects and schedules jobs for the print shop. Figure 2 shows LDP's two-level architecture in which jobs submitted to the shop are distributed to one or more "mini-shops" called *cells* for production (Rai and Viassolo 2003). This is a radical departure from traditional department-style print shop designs in which equipment performing the same or similar functions is clustered together (e.g., all the printers reside in a "printers-only" area, all the finishing devices in a "finishers-only" area, and so on). In contrast, a cell in LDP normally contains a diverse set of machines that can perform different functions.

LDP's cellular design methodology effectively unites various machines and human operators involved in different stages of document production to significantly boost print-shop productivity. The reason is that a cellular design allows each cell to be optimized separately for a better match of the characteristics of the print jobs to the production capabilities of the cell. As a beneficial side effect, work-in-progress (WIP) in LDP-enhanced shops is usually much lower than traditional shops, thanks to the close proximity of the machines residing in the same cell.

Figure 3 shows the block diagram of LDP's scheduling system, which takes as inputs a shop definition file and a jobs definition file.

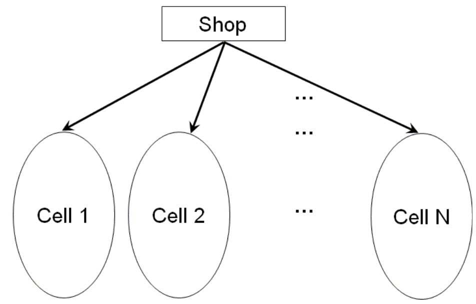

Figure 2: LDP's two-level architecture.

Shop definition: The shop definition file contains information regarding all aspects of the shop, which most closely resembles the domain (i.e., operators) file in PDDL planning. More specifically, a shop comprises of

- *Schedule*: This is the shop-level schedule that describes the operational hours of the entire shop (e.g., 8am $\sim$ 5pm everyday, except for weekends), which can be used as the default schedule for the machines and human operators in the shop.
- *Sequencing policy*: The sequencing policy determines the order in which the jobs are scheduled. Currently, the system supports the following policies: (1) first in first out, (2) earliest due, (3) least slack, and (4) minimum processing time. Of course, more policies can be added, which can be done easily with the current implementation.
- *Machines*: Each machine is identified with a unique name, and is capable of performing a set of function sequences with various speeds, setup times (which may depend on the attributes of the previous job), and pricing information. Optionally, each machine can have its own schedule (e.g., scheduled maintenance between 3$\sim$4pm on Thursday), which overrides the default shop-level schedule.
- *Operators*: Each human operator is identified with a unique name and possesses a set of skills for performing manual steps (e.g., inspection) and supervising machine operations, which are identified by the names of the machine function sequences the operator knows how to operate (e.g., color printing on a continuous-feed printer). In addition, each operator can have his or her own schedule (e.g., working from 9am to 4pm Monday through Friday) that overrides the default shop-level schedule.
- *Cells*: Each cell is made up of a group of human operators and a list of machines they operate or supervise. Individual cells can enable or disable batch splitting, which allows the division of a big job into smaller pieces called *batches*, to further improve throughput. For scheduling flexibility, each cell can choose to ignore the schedules of its operators when assigning jobs to machines, assuming the production schedule is machine-bound instead of operator-bound. The default scheduling option, however, requires that the schedules of both the machines and the human operators be taken into account.

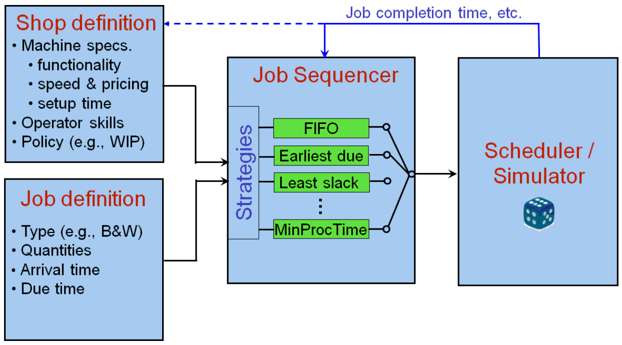

Figure 3: LDP scheduling system overview.

Jobs definition: The jobs definition file contains detailed information regarding the list of jobs submitted to the shop, which most closely resembles the problem instance (i.e., facts) file in PDDL planning. More specifically, a job comprises of

- *Temporal constraints*: The arrival and due dates of a job are provided when the job is submitted.
- *Resources*: Each resource is identified by a unique name (within the job) and describes the quantities required to complete a particular document-production step (e.g., 100 color pages must be printed for the color-print step)
- *Function steps*: Each function step is a unit operation (e.g., color printing on letter-sized paper), which has a (possibly empty) set of input resources and produces a (possibly empty) set of output resources. This is similar to the precondition and effects in STRIPS, except that the duration of each step is non-uniform, depending on the quantities defined in the resources. The steps themselves do not have to form a linear sequence of actions, as in sequential planning. Parallel steps are quite common in print-shop operations (e.g,. the front, the body, and the back matter of a book can be produced simultaneously). Each step can have a set of attributes, which are used to compute the setup time as follows: for each attribute that is changed (compared to the attribute of the last job that was processed on the same machine), a corresponding penalty is added to the setup time.

To schedule multiple jobs, LDP first invokes a job sequencer that orders these jobs according to one of the four sequencing policies shown in the middle of Figure 3. Jobs appearing earlier in the sequence are scheduled before those that appear later. Once the sequence of jobs has been decided, they are fed to a scheduler one at a time such that each subsequently scheduled job must respect all the constraints imposed by those jobs that were scheduled before it. This is obviously a greedy policy that does not guarantee global optimality. However, in practice we have found it works reasonably well in various print shop configurations and job mixes. When faced with multiple scheduling choices, the scheduler always picks the one that finishes the last job as early as possible, essentially favoring plans with a shorter makespan. Note that a similar greedy-search strategy was also used in the Tightly Integrated Parallel Printing (TIPP) project that the PARC researchers have worked on previously (Do *et al.* 2008; Ruml *et al.* 2011).

Upon the completion of the last job in the sequence, the scheduler returns a number of statistics designed to summarize the quality of the schedules found. These statistics include average processing and job turnaround times, average and maximum lateness, and the number of late jobs, among others. They can be used as part of a feedback loop (as indicated by the dashed line in Figure 3) to improve the layout of a shop, because as the job mix changes, so must a cellular design for the shop. In an earlier implementation, the toolkit uses a stochastic simulator to generate the performance statistics, which can vary slightly from one run to another. The search-based scheduler developed by us is the first deterministic scheduler for LDP. From a practical implementation viewpoint, having a deterministic scheduler makes it easy to operationalize our toolkit, since the resulting schedules are free of any idiosyncracy produced by a particular run of the scheduler.

Advanced Features in LDP Scheduling

We next describe a number of advanced features that are first made available in our deterministic scheduler.

- *Adaptive batch splitting*: An important throughput-enhancement strategy in LDP is batch splitting, which chops a large job into a number of smaller units called “batches.” The idea is to eliminate downstream waiting as soon as some portion of a long job is ready for further processing. In an earlier version, the batch size is calculated using fixed formulae that do not adapt to the dynamic workload of each cell. Later on, we designed a fully adaptive strategy that first sub-divides a long job into sufficiently many batches, followed by a merge phase in which the algorithm recursively combines two batches that can be scheduled back to back on the same machine, to ensure only a minimal number of batches are created.
- *Multi-site scheduling*: We extended our basic single-site scheduler to a distributed production environment in which multiple geographically separated sites can efficiently coordinate with one another to share the workload while respecting all their individual resource and sequencing policy constraints. It takes into account the transportation delays between multiple sites when making scheduling choices. Our computational results show huge reductions in the number of late jobs and average turn around time compared to the single-site equivalent that treats each site as an isolated shop. Multi-site scheduling enables better resource utilization and cost reduction, currently a popular trend in the printing industry.
- *Multi-core parallelization*: To support efficient parallelization of the scheduler on modern processors, We developed a multi-core version of our scheduler that uses shared-memory parallelization (based on POSIX threads) to achieve near linear speedup in the number of processor cores used. This is particularly beneficial for multi-site

scheduling, which typically deals with thousands of jobs and a number of cells in each site. Compared to the earlier versions (implemented in Java), our latest C++ implementation is not only much faster as measured in wall-clock times, it is also significantly more memory-efficient. This allows our scheduler to handle larger shops with many more jobs than its predecessors.

Lessons Learned

There are a number of valuable lessons that we learned in the process of developing the LDP scheduler. We highlight a few below.

First, we found that real-world data of print shops and jobs typically contain a great deal of noises such as inconsistent or missing fields. As a result, we had to spend a lot of time on data cleaning and consistency checking to make sure they accurately model the real print-shop environments and jobs. To mitigate such a laborious and error-prone task, we developed an automated consistency checking tool and embedded it inside our scheduler. The tool has been invaluable to us, as it uncovered a number of modeling bugs in existing LDP scenarios that were supposed to be already cleaned.

Second, a well-designed GUI can be crucial to customer adoption and can have a significant impact on the overall productivity of the system. With an earlier version of the toolkit, the feedback we received from the customers was that it could take a while to use all the LDP functionalities and the learning curve was somewhat steep. In the later releases, efforts have been made to simplify and streamline the GUI to make it more accessible to average users. This is very well received by the customers. Figure 4 shows a sample screenshot of the LDP toolkit in its shop definition mode. As shown in the left panel, the toolkit includes the *Job Editor*, *Scheduling*, *Reporting*, *Simulation*, *Monitoring*, *Job Factory*, as well as other tools.

Third, we found it interesting that although the modeling language used by LDP does not resemble much of a "domain-independent" planning language used in the research community, it is actually quite adequate for the print-shop scheduling applications. More surprisingly, it appears that this somewhat domain-specific language can be extended (at a reasonable cost) to other scheduling applications beyond printing, such as the generic job-shop scheduling problems (Pinedo and Chao 1998) found in many other industries. This experience has taught us a lesson on where to strike a good balance between domain independence and scheduler efficiency. From the end user's perspective, which planning language to use is likely a low-visibility issue, since all they interact with is the GUI, and whether it is PDDL or LDP's XML-based, domain-specific language under the hood is of little concern to the end user.

Conclusion & Future Work

We have presented a real-world print shop productivity enhancement tool called LDP that has generated significant revenues for Xerox and its customers. It has roots in cellular manufacturing seen in the automobile industry yet its solutions have all been successfully adapted for and validated by

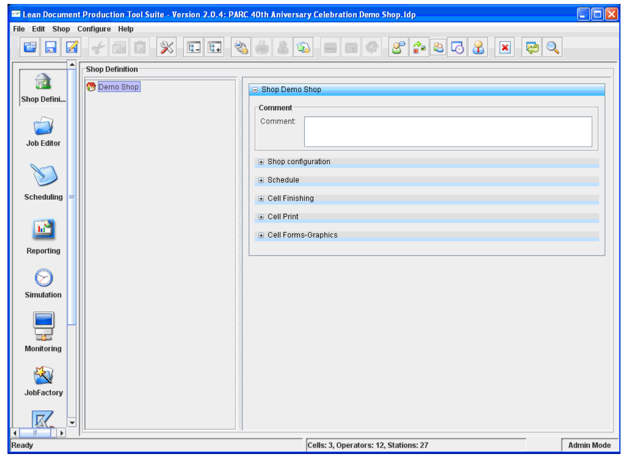

Figure 4: LDP toolkit screenshot.

the printing industry.

In the future, we plan to take the same approach and practice in lean manufacturing, as embodied by our LDP toolkit, to other application domains with similar characteristics. We believe our experience in bringing simple yet effective scheduling techniques to realistic production systems has values beyond the document-production world.

References

Minh B. Do, Wheeler Ruml, and Rong Zhou. Planning for modular printers: Beyond productivity. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS-08)*, pages 68–75, 2008.

Michael Pinedo and Xiuli Chao. *Operations Scheduling with Applications in Manufacturing & Services*. McGraw-Hill, Boston, 1998.

Sudhendu Rai and Daniel E. Viassolo. A lean document production controller for printshop management. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 4, pages 4115–4125, 2003.

Sudhendu Rai, Charles B. Duke, Vaughn Lowe, Cyndi Quan-Trotter, and Thomas Scheermesser. LDP lean document production – O.R.-enhanced productivity improvements for the printing industry. *Interfaces*, 39(1):69–90, 2009.

Wheeler Ruml, Minh B. Do, Rong Zhou, and Markus P.J. Fromherz. On-line planning and scheduling: An application to controlling modular printers. *Journal of Artificial Intelligence Research*, 40:415–468, 2011.