BETTER BUSINESS ANALYTICS

COMPLEX PRODUCTION SCHEDULING: METHODOLOGY AND CASE STUDIES

Introduction

Current ***market forces*** are putting ***pressure on manufacturers*** to ***increase flexibility and customization.*** Whether production is ***make-to-order or make-to-stock,*** the trend is towards ***shorter production*** runs and more ***frequent product changeovers,*** which increases the need for ***better production scheduling*** capabilities.

Companies whose ***production costs*** represent a ***significant portion of the price*** of their products can gain a source of ***competitive advantage*** by creating ***optimal production schedules.*** Complex manufacturing operations where multiple products often share ***common infrastructure and resources*** require ***production schedules*** on a timely and frequent basis. In addition, plant expansions are very expensive; ***better production schedules*** often allow companies to ***increase throughput*** without incurring large capital expenditures, resulting in ***increased product gross margins.***

ERP, ***production planning,*** and several dedicated commercial ***off-the-shelf (COTS)*** solutions exist for ***production scheduling.*** However, these solutions ***i)*** only provide reasonable schedules at a coarse, weekly or monthly level, ***ii)*** produce schedules that are ***not optimal*** and/or require ***ineffective manual manipulation*** and/or ***iii)*** cannot be re-run daily, or ad-hoc, to address ***immediate issues and priorities.*** Further, while these solutions produce ***working schedules,*** they often ***fail to accommodate related constraints*** that can only be factored in by using a schedule evaluation or simulation solution. Examples of such constraints include ***WIP inventory and related storage limitations,*** sequencing conflicts on common machines, or resource-sharing restrictions.

The vast ***production planning and production scheduling*** literature includes many approaches based on ***pure optimization,*** simulation, and ***hybrid simulation-optimization methods.*** Recent surveys identify numerous implementations ***involving discrete event simulation approaches,*** with more than a dozen others involving ***alternative types of simulation.*** Numerous additional implementations involve ***complex mathematics and metaheuristics*** (i.e. designs or strategies to ***efficiently explore all options*** in order to find near–optimal solutions). However, these diverse implementations ***share a common denominator*** of exhibiting one of three conspicuous limitations: either they make ***severe simplifications of the processes*** they are trying to represent, or they only ***consider portions of a complete process,*** or they do not provide an integrated ***system capacity*** and job sequencing framework.

Existing solution approaches for ***production process scheduling*** characteristically focus on two basic questions:

- When should a ***specific job be scheduled?***
- What ***resources*** should be ***assigned*** to perform the job?

In many cases, these questions can be answered by ***applying simple rule-based mechanisms,*** such as ***sequencing tasks by Earliest-Due-Date (EDD)*** or by the magnitude of their processing times. ***More complex*** rules can be derived by ***combining two or more simple rules*** into ratios or products, but the ***basic concept remains the same.***

Although appealing for their ***simplicity and intuitive nature,*** these methods usually produce ***inferior results*** because they are static in nature and tend to ***ignore relevant attributes of the tasks,*** such as urgency to begin production, ***penalties for tardiness,*** interactions with other tasks, ***availability of resources to perform all the work,*** changeover and setup times and costs, etc. To address these limitations, optimization-based approaches can be used. These methods use ***mathematical programming techniques*** to find an ***optimal solution*** to maximize or minimize some metric, such as throughput, ***capacity utilization,*** makespan, or operating cost.

The ***complexity of most real-world systems*** involves ***several decisions,*** including:

- How to size a task ***(i.e., job, batch, run, etc.)***
- How to ***assign a task*** to a production line
- How to sequence the tasks ***on each production line***

Unfortunately, in ***high product mix environments,*** where product changeovers are ***sequence-dependent,*** the application of ***exact mathematical optimization methods*** is impractical, either because the ***time*** to obtain the ***optimal solution*** is ***excessive,*** or because such systems are ***too complex to be mathematically formulated.*** In these cases, what is needed is a ***combination of mathematical methods with metaheuristic solution techniques*** and, increasingly, simulation ***modeling approaches.***

In response to this need, **OptPro** from **OptTek** provides a ***sophisticated production scheduling solution*** approach that ***combines mathematical programming,*** metaheuristic ***optimization,*** and ***simulation*** to craft ***optimal or near-optimal production schedules*** in a timely, ***reliable,*** and effective manner.

The modeling and ***algorithmic designs*** employed are especially suited to ***complex situations,*** with a ***high production volume*** and a high product mix, where ***sequence-dependent constraints,*** multiple line assignments, ***scarce resources,*** and ***tight storage*** and ***WIP constraints*** may be present. Such a ***design structure*** also proves ***very effective*** when a major disruption occurs in the plant, and there is an ***urgent need*** to quickly reoptimize the ***production schedule.***

OptPro Methodology

The **OptPro** production scheduling approach makes use of ***multiple technologies,*** either alone or in combination, ***tailored to the situation at hand.*** What every implementation has in common, regardless of the ***individual technologies*** employed, is a ***technological framework that coordinates and unifies*** the function of its components. This framework can be described as a ***scheduling optimization engine,*** which draws on a ***diverse set of techniques*** to obtain an ***optimal or near-optimal production schedule.*** These techniques include ***mathematical programming,*** metaheuristics, and the combination of ***simulation and optimization.*** **Figure 1** shows a ***high-level representation*** of the technology.

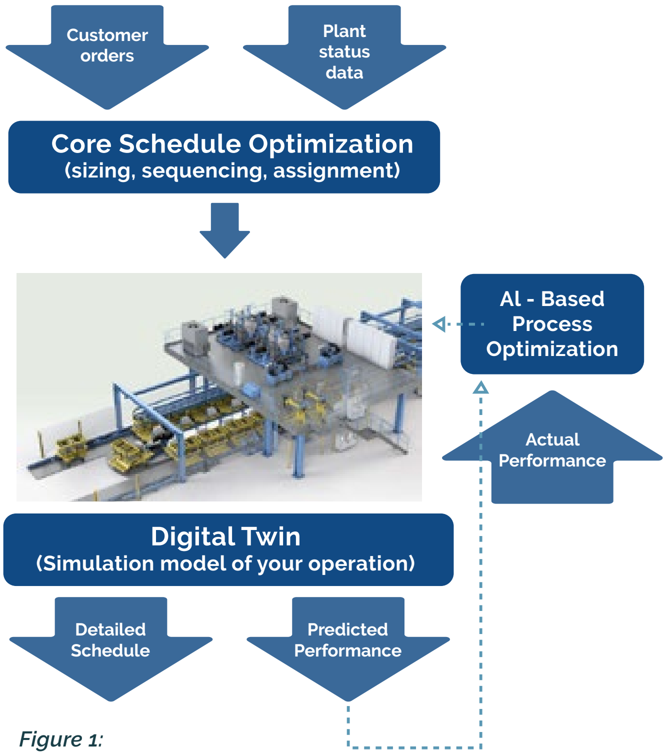
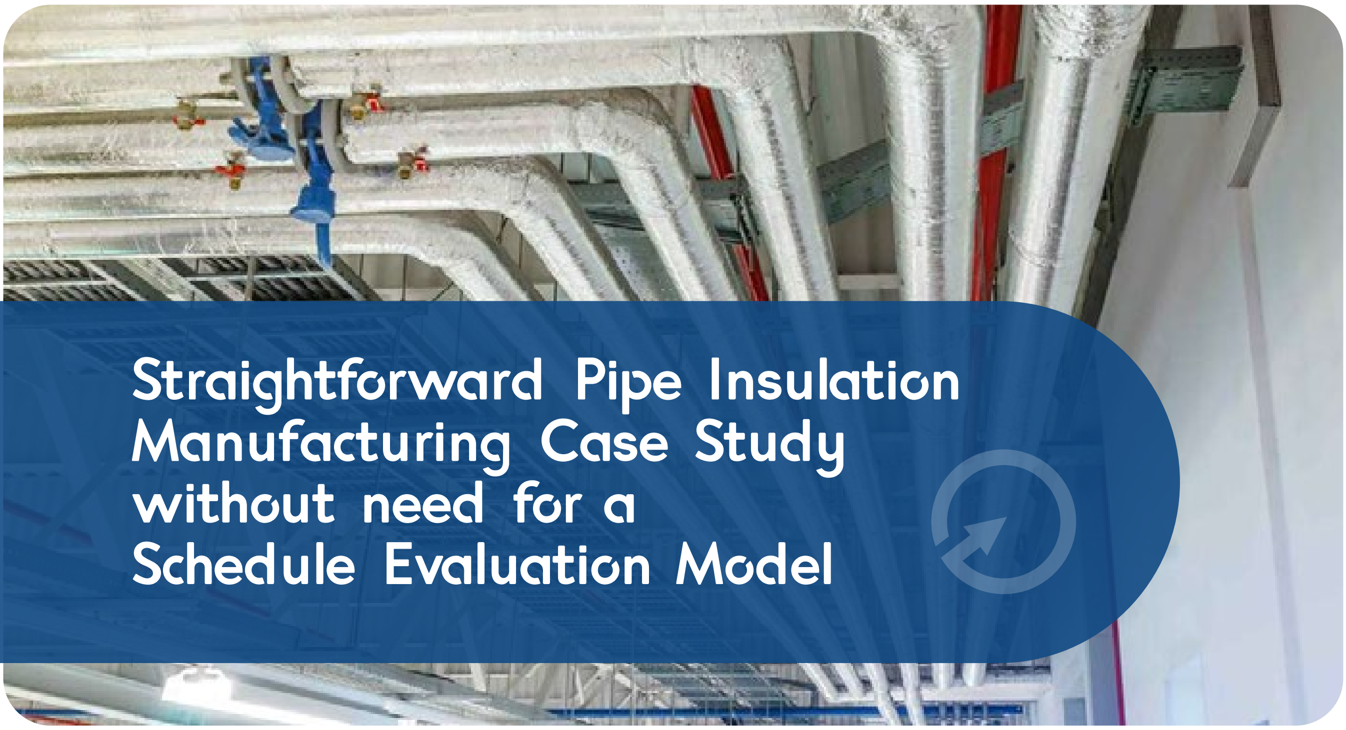


Figure 1:
High-level representation of the technology behind ***OptPro***

Even ***highly complex production environments*** can sometimes be ***tractable enough*** that they can be ***solved*** with the ***Core Schedule Optimization*** alone. Examples are operations where ***production lines are independent*** of each other (i.e., there is little or no interaction ***between production lines),*** and resources such as labor and physical assets are primarily ***dedicated to each production line.*** Most real-world systems, however, exhibit a degree of ***complexity and interactivity*** that makes it impossible to describe them with a set of ***equations or mathematical expressions.*** A good example of such a system is a ***dairy foods production facility.*** While it may be relatively simple to ***optimize the schedule*** of the filler machines and the packaging facility separately, ***the schedules produced*** without accounting for their interactions may ***introduce conflicts in the upstream operations*** of the process. Good coordination between ***processing and packaging is critical,*** so that the ***scheduling*** of the pasteurization step and the sterilization step, for example, are ***well-synchronized with the scheduling of the filling and packaging operations.*** To handle crucial interacting factors such as these, **OptPro** makes use of a ***simulation model*** called the **Schedule Evaluation Model (SEM)** – a ***"digital twin" of the process.*** The **SEM** is a realistic model of the ***production facility,*** which allows an iterative ***evaluation of schedules*** suggested by the ***optimization engine,*** in a ***simulation optimization*** sense. It is important to note that the **SEM** is only as detailed as is necessary to model the effects of ***scheduling on a production process.***

Again, this approach is not needed in situations where ***straight-forward rules*** such as ***First-In-First-Out (FIFO)*** or ***Earliest-Due-Date (EDD)*** would suffice. Instead, it is ***designed for those operations*** where, as noted, ***multiple products*** compete for ***common resources,*** such as production ***infrastructure and materials,*** and where an ***optimal production schedule*** can drive ***competitive advantage.*** In general, this approach ***finds application in organizations*** that seek to ***optimize and automate*** their plant design and production schedule, or to ***maximize the benefit*** derived from their ***operational processing decisions.*** In manufacturing settings, ***improved decision*** making is enabled by ***simultaneously optimizing scheduling,*** sequencing, line-assignment, capacity, and layout decisions to meet forecasted ***customer demands.***

The following ***case studies*** highlight the ***benefits*** of the **OptPro** approach.



Straightforward Pipe Insulation Manufacturing Case Study without need for a Schedule Evaluation Model

Consider a ***large construction materials company*** that produces ***pipe insulation products*** in one of its manufacturing facilities.

Prior to implementing **OptPro,** the company was ***employing an advanced planning and scheduling (APS) system*** that used a set of ***dispatch rules.*** While the system provided some ***user-friendly capabilities,*** the ***complexity of the operation*** had grown in recent years to the point that the ***current system*** was producing schedules that required ***hours of manual "fixing"*** to make them feasible.

The company ***produces over a hundred different pipe insulation products.*** Each product is described by the ***type of material,*** the internal diameter, and the thickness of the insulation, and is ***manufactured on one or more of several available production lines.***

The ***production facility*** works on a ***24/7 schedule*** and the company's aim is to ***schedule 30 days of production at a time,*** with a detailed plan for the first ***5-7 days*** and a ***less granular plan*** for the remaining ***23-25 days.*** In this example, based on the ***current conditions,*** a ***schedule is required for a single day of production*** for five products on three available ***production lines.***

The ***schedule needs*** to define:

◦ How much of each product must be ***produced each day*** on each line – i.e., what production ***run size,*** while ***maximizing*** total throughput

◦ What is the ***best sequence of product*** runs on each line to ***minimize changeovers*** and avoid overtime?

The ***production requirements*** for the day are shown as follows in **Table 1:**

SKU	Processing Rate (lbs/minute)	Possible Machine Assignments	Day's Demand (lbs)	Safety Stock Shortfall (Surplus)
1	150	1, 2	66,000	8,000
2	150	2, 3	27,000	0
3	75	1, 2, 3	90,000	4,500
4	75	1, 2, 3	48,000	(5,000)
5	75	2, 3	60,000	0

Table 1: Production requirements

In this case, the ***quantity to produce*** each SKU is the ***Day's Demand*** plus any shortfall (or, minus any surplus) in ***safety stock***. The ***total quantity to be produced*** must, therefore, meet or exceed ***298,500 pounds*** of finished product ***(the sum of the five SKU's Day's Demand).***

Additionally, the sequence-dependent ***changeover times*** incurred when ***changing a machine*** from production of ***one product to another,*** in minutes is as follows in **Table 2:**

Product	1	2	3	4	5
1		60	60	60	60
2	60		60	60	180
3	180	120		60	120
4	180	180	120		120
5	60	180	120	120	

Table 2: changeover times

To ***avoid paying its operators for overtime,*** the plant must ***finish production within 20 hours.*** The additional 4 hours available at the end of the last shift are typically used for ***preventative maintenance*** and cleaning activities.

Optimal production schedule for Day 1

The optimal schedule is as follows in **figure 2:**

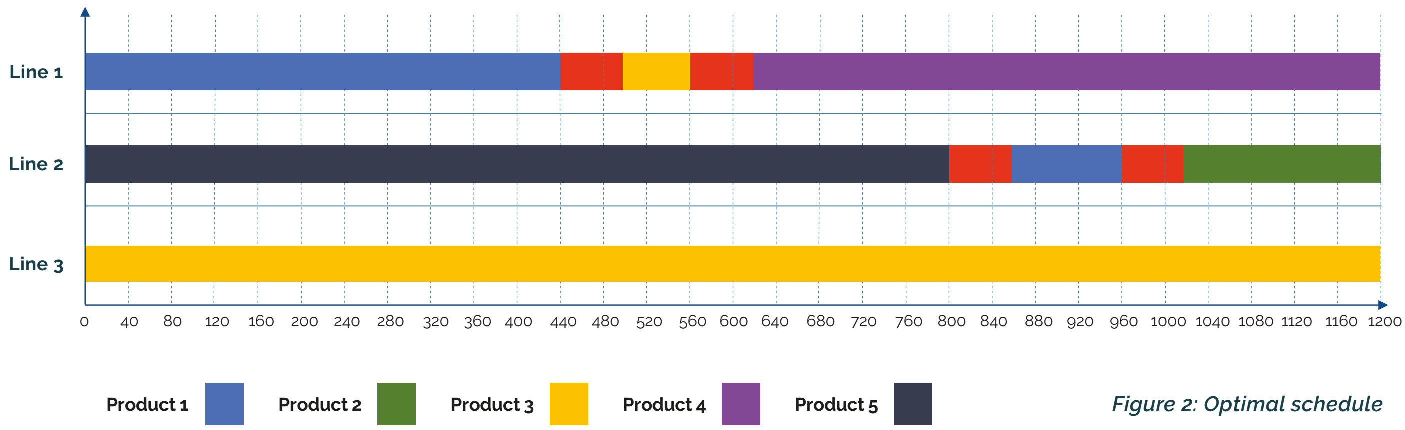


Figure 2: Optimal schedule

Perhaps not intuitively, but ***optimally,*** both Products 1 and 3 are ***split into two batches,*** to be ***produced on different machines*** to avoid incurring ***overtime costs*** (one of the key objectives). The ***schedule maximizes throughput,*** resulting in a total production of 306,500 pounds, and ***minimizes costs,*** including changeover costs ***(240 minutes).***

Rapid Re-Optimization

However, as a result of unforeseen ***tool breakages,*** the scheduler realized this ***schedule cannot be put in practice*** because of ***tooling conflicts*** in resource utilization; two batches of the same product cannot now be run in parallel ***(Product 3 on Lines 1 and 3),*** because they each require a mandrel of the same diameter and there is only one. In addition, ***Products 2 and 4 also share a mandrel,*** so those two cannot run in parallel either ***(on Lines 2 and 3).***

The ***appropriate tool constraints*** are easily edited to ***reflect shared resource conflicts,*** the optimizer is very quickly re-run ***(a matter of minutes),*** and a new ***optimal solution*** is obtained as follows in **figure 3:**

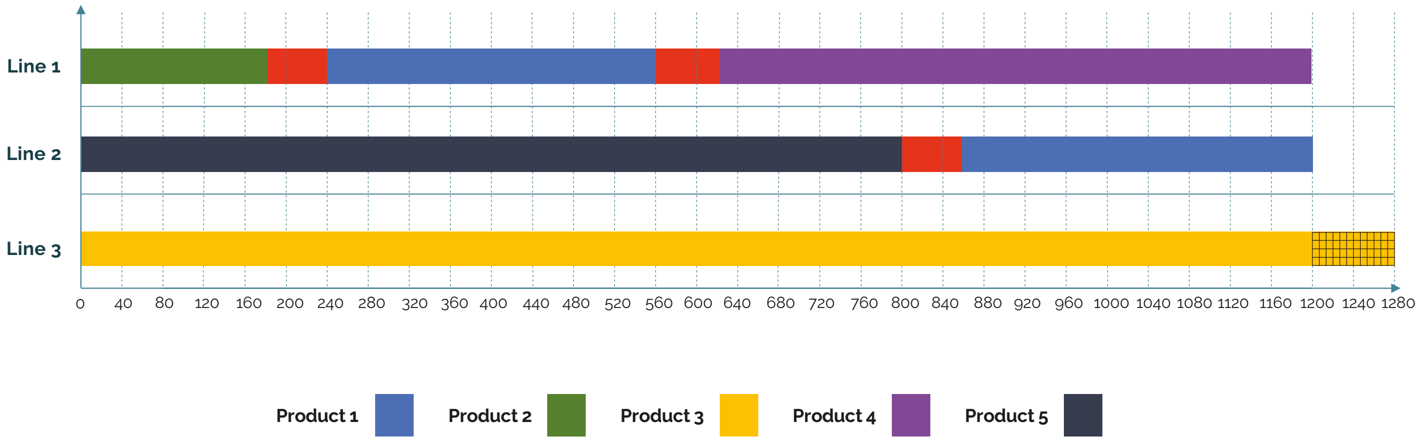


Figure 3: Optimal schedule after rapid re-optimization

This ***new solution avoids*** the ***tool conflicts mentioned above.*** Although this new solution does now incur 60 minutes of overtime on Line 3, it ***reduces the total changeover time by 60 minutes,*** practically ***offsetting the cost of overtime*** with ***revenues*** realized from additional throughput of 18,000 pounds of ***finished Product 4.***

Given this is a ***real-world example,*** the approach illustrated here resulted in an ***8% increase in throughput compared to a manual approach*** being used beforehand, with an overall **10% cost savings** from changeovers and still ***reduced overtime.***

DAIRY INDUSTRY CASE STUDY
WITH A SCHEDULE EVALUATION MODEL

OptPro was implemented for a ***dairy processing and packaging plant*** design and equipment manufacturer, with an ***implementation carried out at a medium-sized dairy production facility*** with one pasteurizer/separator, one sterilizer and six filling machines ***(a simplified schematic of the facility is provided in Figure 4).*** The facility had a ***difficult time creating a schedule*** that would produce ***enough product to meet demand,*** without the need for ***overtime*** and without the current ***corner-cutting on cleaning and maintenance.***

The ***challenge*** was to create a ***schedule*** that would:

- ***Minimize the time*** to produce SKUs – keeping ***below 100% indicating no need for overtime,*** and ***allowing time for cleaning & maintenance***

- ***Maximize the existing utilization,*** but keeping below 100% indicating no need for additional capacity/infrastructure

- Produce a schedule, and ***be able to reschedule,*** quickly

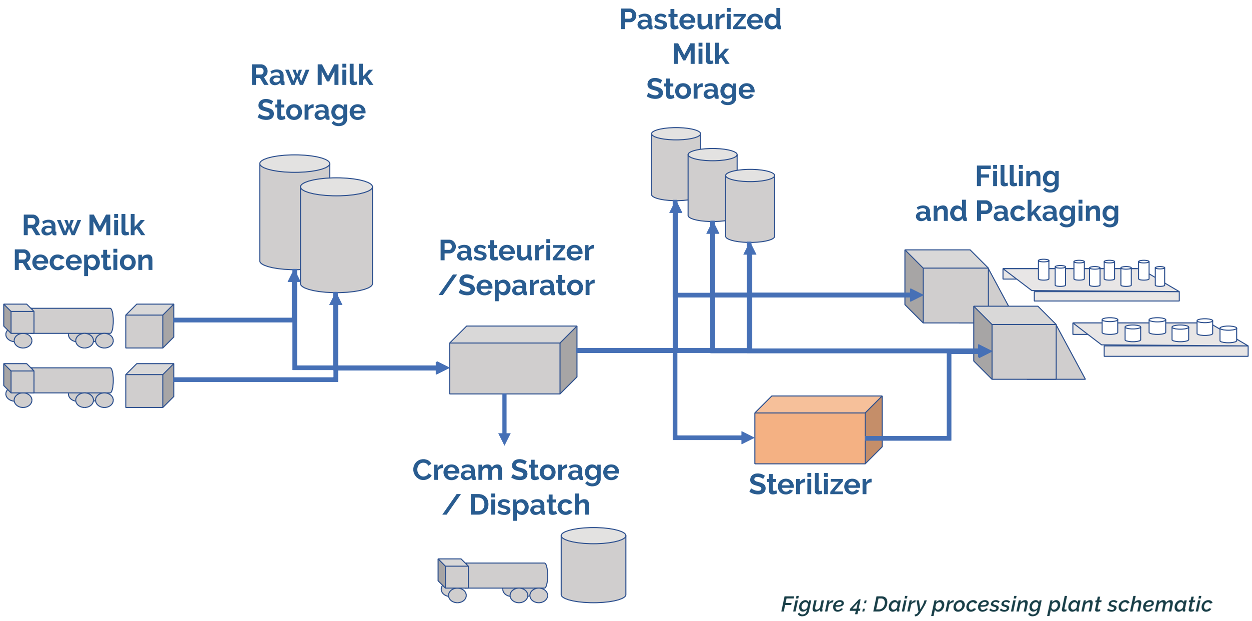


Figure 4: Dairy processing plant schematic

Demand is specified by ***17 SKUs of different dairy product types and package sizes.*** In the plant, ***raw milk reception occurs daily,*** with raw milk storage silos ***directly filled by pipes*** from the reception area. Siloes are then ***pumped through to the pasteurizer/separator*** where the ***raw milk is separated into pasteurized milk*** with different fat content specifications, and a cream byproduct. ***The cream is stored until it can be dispatched*** at the end of the day. ***Pasteurized milk is stored until a filling machine is available,*** and then it is either filled as ***fresh product,*** mixed with flavoring and filled as flavored product, or sterilized and ***filled as ultra-high temperature (UHT) product.*** In the final step of the process, filling machines fill containers of a prespecified size with the appropriate final product. ***A SKU is defined by the combination of finished product type and container size.***

OptPro first applied a ***sequencing and assignment algorithm to the process*** at the filling step. ***Four distinct tests were then executed,*** and the ***time was measured for completion,*** along with the ***quality of the solution*** in terms of makespan and equipment utilization. ***The tests and their respective quality results are described in* Table 3.**

#	Type	Obj	Time	Mksp	Util
1	A	U	7s	99.1%	95%
2	AE	U	15s	99.1%	96%
3	M	M	11s	97.4%	85%
4	ME	M	9s	97.4%	85%

Table 3: Evaluation tests

In Column 2 of Table 3, the "Type" of test is defined as follows:

A: *Average:* → weekly production is averaged over 7 days, such that one-seventh of the weekly demand must be completed in each day.

M: ***Makespan minimizing schedule*** → total weekly production ***must be completed*** as soon as possible.

E: *Expanded* → the ***number of SKUs is doubled to 34,*** but the ***demand per SKU is halved,*** to test the ***flexibility of the algorithm to larger quantities of SKUs,*** while ***guaranteeing that a feasible schedule exists.***

Thus, ***if Column 2 contains the abbreviation:* "AE,"** it means that the test is of type *"Average, Expanded".* This ***means that the test involves a daily average demand*** equal to one-seventh of the ***total weekly demand for each SKU,*** and the number of SKUs will be 34, ***but the demand for a SKU will be half that of the original SKU.***

Column 3 shows the primary objective(s) of the test case, either ***U = equipment utilization (maximized) or M = makespan (minimized).*** Time, ***in Column 4, reflects the computer run time*** – in seconds – to obtain the ***best solution. Columns 5 and 6*** display the ***resulting values of Makespan and Utilization,*** respectively. These are all expressed as percentages. Thus, makespan is reported as a percentage of the ***maximum allowable production time*** (i.e., 1,440 minutes per day for ***tests involving an "Average" production requirement,*** and 10,080 minutes per week for ***tests involving a "Makespan-minimizing" production requirement).***

The ***results indicate*** that ***all schedule options were viable,*** and all executed quickly. None showed ***Utilization > 100%,*** indicating ***no need for additional capacity.*** All showed ***Makespan < 100%,*** indicating ***no need for overtime,*** and leaving time for ***cleaning and maintenance.***

Need for Schedule Evaluation Model or SEM (simulation)

It currently takes an ***experienced process engineer*** over five hours to obtain a ***good scheduling solution for the week.*** Part of the reason is that a ***filler schedule must consider*** the coordination ***between the filler schedule and the activities*** at the upstream equipment – namely the sterilizer and the pasteurizer. If these are not in ***step with the fillers,*** then the filler ***schedule can be greatly disrupted*** by long idle times or costly and ***untimely changeovers.*** Although **OptPro** produced ***excellent solutions for the fillers in a very short computational time,*** it was necessary to ***verify that the upstream equipment could also be schedule***d in a way that ***minimized disruption of the fillers' schedule.*** This proved ***challenging,*** especially in the case of the ***sterilizer whose operation is more constrained in terms of buffer capacity*** - whereas the pasteurizer relies on large, relatively ***non-expensive storage tanks,*** the sterilized product must either be ***fed directly to the filler,*** or it must be ***stored in relatively small, expensive aseptic tanks.***

To address this issue, a detailed **SEM** of the dairy plant ***was modeled.*** After initial testing, it was ***necessary to modify the initial solution generator*** in the **OptPro** ***algorithm to eliminate much of the reliance on randomization.*** The ***new solution construction method*** then ***selected SKU sizes*** and filler machine assignments in close coordination with the sterilizer to ***avoid periods of idle time at the fillers,*** ensure ***continuous operation of the sterilizer,*** and ***minimize changeovers.*** This new approach was ***successful*** and although the ***total run time*** for each of the test cases described earlier ***(see Table 1)*** now took ***between 1 and 4 minutes,*** the ***schedule options were again all viable*** and this increase in time still represents a ***great improvement compared*** to the ***5 hours*** it currently takes ***the engineer.***

A High-Volume eCommerce Printshop Operation with Schedule Evaluation Model

The ***final case*** refers to a ***high-volume printing operation.*** Customers ***upload a design*** and then ***order products*** such as ***photobooks, calendars, and other personalized items*** from the company's ***online store.*** The company receives ***several thousands of orders each day*** during peak periods. Each different item in an order is ***assigned a production ticket ID.*** Thus, some orders may be ***associated with a single production ticket,*** and some with ***several tickets.*** On average, the company ***generates over 50 thousand production tickets per day.***

The ***company provides*** the customer with a ***planned ship date for each order,*** determined by the ***type and number of associated production tickets.*** For example, an order that contains ***multiple and complex types of items*** may be assigned an ***expected ship date*** that corresponds to ***5 workdays after the order was received,*** whereas an order with a ***single, and simple,*** item might be assigned a ***ship date*** corresponding to only ***2 days after the order was received.*** The ***primary goal*** of the company is to ***maximize on-time-shipment,*** measured as the percentage of orders that ***ship on or before the planned ship date.***

Current ***scheduling is, to a large extent,*** performed following a ***First-Come-First-Serve (FCFS) approach.*** The only ***"intelligence"*** added to ***FCFS*** is that for a ***given backlog of production tickets,*** similar items are ***processed together as batches,*** to minimize changeovers. By and large, the biggest ***bottleneck in the process is the printing step*** which occurs first for ***most products.*** This step involves both the ***longest processing times*** and the ***longest changeover times*** of any step in the process. As a result, ***jobs can be sequenced at the printers,*** ignoring any of the steps occurring downstream - the problem is then effectively reduced to a sequencing problem. A ***Schedule Evaluation Model*** **(SEM)** of the plant is ***used to simulate the optimal sequence*** and collect detailed metrics. The **SEM** models each ***step of the process.***

A ***high-level process*** of the plant is depicted in the ***flow chart of Figure 5.*** The multiple machine sequencing problem with ***sequence-dependent setups*** can likewise be ***formulated with an objective*** that can be expressed as ***minimizing the number of tardy jobs,*** such that a ***complete formulation*** can be computed.

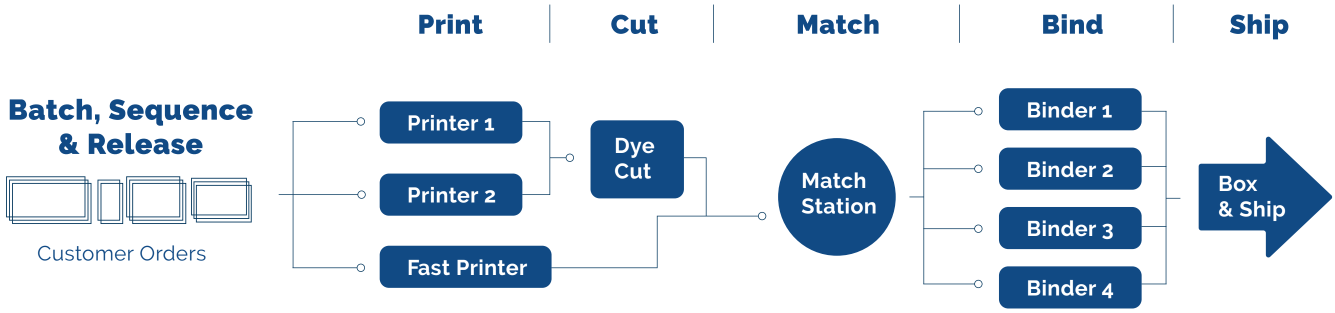


Figure 5: High level flow chart of printing process

The ***solution approach*** involves a ***unique search procedure*** with a ***"greedy" heuristic construction method*** (i.e., the ***locally optimal choice*** is made at each stage) and ***efficient neighborhood search.*** The greedy constructor is ***based on the Apparent Tardiness Cost with Setups (ATCS) rule,*** applied to the ***total weighted tardiness*** on a single machine where ***jobs are sequenced*** in descending order of a priority index.

To illustrate this method, ***consider a set of 5 jobs*** as shown in **Table 4.**

Job	Proc. Time	Weight	Due Date
1	30	7	80
2	40	8	100
3	10	2	120
4	40	3	170
5	50	5	190

Table 4: Jobs data

Setup times for these products are shown in **Table 5.**

Setup Time	1	2	3	4	5
0	9	6	8	8	12
1		13	7	12	11
2	9		11	13	6
3	9	10		20	7
4	10	7	8		6
5	14	13	12	13	

Table 5: Setup times, including initial setup

The priority ***indexing calculations and resulting sequence*** for jobs yet to be selected at each time interval are shown in **Table 6.**

t	1	2	3	4	5	Sequence
0	0.01565	0.02822	0.01231	0.00410	0.00177	2
46	0.02051		0.00723	0.00141	0.01155	1
85			0.02653	0.00232	0.00381	3
102				0.00030	0.01229	5
159				0.00231		4

Table 6: Applying ATCS indexing to the set of 5 jobs

The total ***weighted tardiness metric*** for this sequence is shown in ***Table 7.*** This method ***compares favorably*** to other ***well-known*** dispatch rules. It outperformed an ***Earliest-Due-Date (EDD)*** method by 21% (total weighted tardiness = 204), ***shortest processing time by 33%*** (total weighted tardiness = 240) and does only ***1% worse than Moore's algorithm*** (total weighted tardiness = 159). The latter is well known as an ***algorithm that minimizes the total number of tardy jobs,*** but it cannot directly address changeovers or weighted tardiness metrics. The fact that ***ATCS does comparably well leads*** us to the conjecture that ATCS is ***well-suited as an initial sequence constructor*** for the case of ***minimizing the total number of tardy jobs*** and weighted tardiness.

Sequence	Setup Time	Start Time	Completion	Tardiness	Weighted tardiness
2	6	6	46	0	0
1	9	55	85	5	35
3	12	92	102	0	0
5	6	109	159	0	0
4	12	172	212	42	126
				Total	**161**

Table 7: Computing total weighted tardiness

Implementation complexity

In our implementation, ***the ATCS rule*** had to be ***modified in two fundamental ways*** in order to: **(1)** ***handle multiple machines;*** and **(2)** appropriately ***time the release of "related" items.*** The first modification is ***obtained by changing the ATCS indexing function*** to represent the time at which the ***first feasible machine will become available.*** (Some machines may ***not be able to process a job,*** so they are not feasible for that job.) The second modification was necessary because of a ***process particularity:*** the company ***consolidated all items in an order into a single shipment.*** That meant that, oftentimes, items with significantly different processing times ***had to be shipped together. In order to limit WIP inventory,*** the release of items in a ***given customer order*** with much ***shorter processing times*** had to be timed in such a way that they would ***finish processing*** at approximately the same time as those with ***longer processing times.***

For this implementation, **OptPro** again used a ***Schedule Evaluation Model*** **(SEM)** to ***evaluate solutions.*** In this case, the **SEM** models the full plant, which ***provides a more precise measure of the performance*** of the system than the ***estimated factors in the ATCS*** (modified to manage multiple machines). This proved essential because ***WIP inventory can build up in different areas of the plant,*** causing ***wait times*** that are not addressed in the ***ATCS indexing rule.***

Due to the ***large volume of incoming orders,*** the procedure is ***repeated at various intervals*** during the day, or whenever a ***major disruption occurs in the plant.*** This is necessary to ***avoid excessively large volumes of backlogged orders,*** which ensures that the ***optimization runs quickly enough,*** and produces ***high-quality results.*** On the other hand, running the procedure too often makes it impossible for ***enough items to be batched into large enough chunks.*** There are certain batch processes in the plant, where ***producing a single item*** requires the ***same processing time*** as processing a batch of 50 items. Thus, the "chunking" step is critical in ***ensuring the efficiency of the solution by creating batches of similar jobs*** (i.e., jobs with characteristics similar enough that there is ***no changeover or setup time*** incurred between them).

Based on ***preliminary testing and simulations*** conducted on historic data, during ***normal months of operation*** (January through September), the **OptPro** implementation typically ***improves on-time shipment of customer orders*** from a current average of ***91% to between 98% and 99%.*** However, the ***solution is most valuable during the peak holiday months*** between October and the end of December, when the current ***on-time shipment metric is only 75%.*** During these periods, the ***solution approach typically yields on-time shipment performance*** between ***92% and 96% (a significant improvement of 23% to 28%).*** These outcomes are ***expressed as ranges due to variability in the results*** at different ***time intervals,*** determined by the ***average complexity and volume of the orders*** in the backlog at different times.

CONCLUSIONS

The ***combination*** of ***custom mathematics, metaheuristic-based algorithms,*** and ***simulation evaluation models*** has proven ***very effective in a wide variety of complex production scheduling*** applications – where ***ERP,*** planning and ***COTS scheduling solutions*** are ***challenged in handling the complexity*** and other requirements.

The ***efficiency and quality of solutions*** obtained by the **OptPro** method makes it ***well-suited for handling problems*** not only in ***strategic production design*** and ***planning situations,*** but in ***real-time, operational scheduling.***

Often, ***effective scheduling solutions*** require a ***combination*** of ***mathematical methods*** with ***metaheuristic solution techniques*** coupled with ***simulation modeling approaches;*** **OptPro's** ability in this regard makes it an ***ideal solution for many complex scheduling challenges.***

OptPro also ***proves useful for re-optimizing a schedule in the event of a major disruption in production*** (e.g., a machine breakdown). This ***re-optimization*** has a ***crucial role*** in many settings, where the ***usual reaction is to continue with the planned schedule*** and ***"work around" the disruption,*** which often leads to ***increased operating costs and excess product waste.*** By enabling the schedule to be ***re-optimized in a matter of seconds,*** the disruption and its estimated duration can be ***addressed directly in the new schedule*** – thus minimizing these negative effects – ***and production can resume immediately.***

BBA

4684 S Hampton Cir
Boulder, CO 80301

720.987.6111

www.bettersolv.com