2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

Press Sheet Optimization for Open Loop Control of Industrial Scale Gang-Run Printing*

Daniel Fullmer and Sean Warnick
Information and Decision Algorithms Laboratories
Brigham Young University
Provo, UT 84602
daniel.fullmer@byu.edu

Abstract— **We consider a problem in the printing industry for gang-run printing of paper products over multiple days. The competitive and high-volume nature of the industry mean that small gains in efficiency translate to significant cost savings and ultimately lower prices. For each day, the problem is to choose which orders to print and the press sheets to print them on that would minimize the total cost of production. We formulate this problem as an integer linear program and find feasible solutions using traditional solvers. This method was compared with optimization by hand in a real factory and significant cost savings were found.**

I. INTRODUCTION

The problem under consideration involves printing of various types of paper products:

$$t \in T,\ T = \{\text{'business cards', '4x6 postcards'}, \dots\}$$

These products are produced on large sheets of paper. The sheets are much larger than any individual product, so many products may be combined, printed simultaneously, and later cut apart. Gang-run printing refers to printing multiple products simultaneously on a single press sheet as opposed to having a press sheet dedicated to each product. This is more cost-effective because the per-run costs of printing are distributed among the products being printed.

There are many possible applications of control and optimization in the printing industry. Previous published work in this field has focused on determining how press sheets should be laid out physically to maximize paper efficiency[1]. At the level of individual presses, most control applications have focused on controlling the physical systems of a printing press[2][3]. Other work focuses on optimizing the workflow and scheduling of jobs in a printing factory[4][5][6]. Still other work focuses on simulation of commercial print production systems[7]. In addition to the published research, commercial products are available that choose and optimize the layout of press sheets. Metrix[8] is one example that optimizes the choices and layout of press sheets based on the cost of production. Additional background information about printing may be found in [9].

We consider printing over multiple days, where new incoming orders need to be printed by some day in the future. On any given day, those orders that are currently due should be printed, while orders that are not currently due may be printed if desired. Additionally printing orders that are due in the future may enable us to more efficiently combine orders onto press sheets. Each day, we choose the set of orders to print and the press sheets to print them on that would minimize the total cost of production. Our approach is novel in considering the cost of production over time, choosing orders to print that may be due in the future, and in our notion of order attributes and supporting press sheets.

Section II explains gang-run printing at industrial scale. Section III presents our problem formulation and solution method. Section IV shows the results of our method using historical data from a real printing factory with a comparison against optimization by hand.

II. GANG-RUN PRINTING

A. Printing Press Technology

There are two main printing technologies used to produce these types of products: digital printing and offset printing. Digital printing typically refers to inkjet or laser printing. These work similarly to printers in the home or office. However, the digital presses used at industrial scale often use much larger paper sizes.

Offset presses transfer ink through a series of cylinders. Paper sheets are run through the press, and the press sheet image is impressed on the paper. Printing plates determine where ink will be deposited (offset) on the paper. New plates must be built for each new job, and require a significant amount of time and money. These plates are then wrapped around a plate cylinder and used internally in a printing press. Each color requires a plate, so, a typical four color job (cyan, magenta, yellow, and black) would need four plates. If a job requires printing on both sides of the sheet, it would require eight plates. Press operators must also ensure that ink is properly distributed across the press sheet. Before the final sheets are actually printed, some sheets are run through the press to bring it "up to color". Switching jobs on an offset press requires a significant amount of time for changing plates, checking colors, and other tasks. The process of setting up a job on a press is called make-ready.

See [10] for a survey comparing digital and offset printing quality. Generally, low quantity jobs will be run on digital presses and high quantity jobs will be run on offset presses. This is because offset presses have significant per-run costs

*This work was supported by Print Fulfillment Services, LLC, 2929 Magazine Street, Louisville, KY 40211.

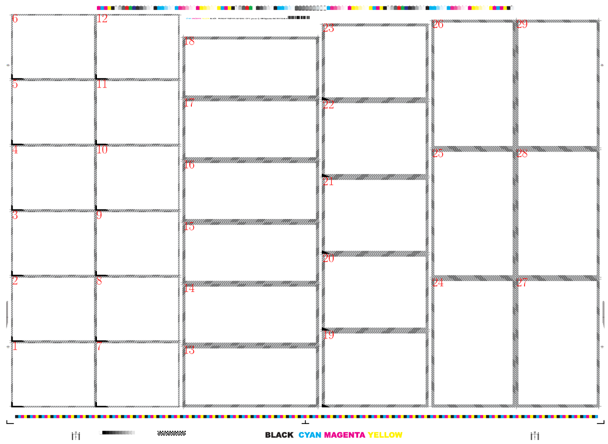

Fig. 1. Example press sheet template with slots for various products

but lower per-sheet costs. We denote the various printing presses in a factory as:

$$m \in M,\ M = \{\text{'Offset Press 1', 'Digital Press 1'}, \dots\}$$

B. Press Sheet Templates

Instead of choosing a new press sheet layout for each press sheet, we use a set of templates that define the layout on a press sheet.

$$p \in P,\ P = \{\text{'18x46PC', '63xBC', '9x46PC-36xBC'}, \dots\}$$

See Fig. 1 for an example of a press sheet template. Each template has a number of available slots where products may be placed, denoted by $u_{pt} \in \mathbb{N}$. If a press sheet does not support a particular product, then $u_{pt} = 0$. For example:

$$u_{\text{'18x46PC','4x6 postcards'}} = 18$$
$$u_{\text{'18x46PC','business cards'}} = 0$$
$$u_{\text{'63xBC','4x6 postcards'}} = 0$$
$$u_{\text{'63xBC','business cards'}} = 63$$
$$u_{\text{'9x46PC-36xBC','4x6 postcards'}} = 9$$
$$u_{\text{'9x46PC-36xBC','business cards'}} = 36$$

This simple set of templates provides a good example of the value of paper efficiency. Given a number of business card and postcard orders, we could produce one template from each of 18x46PC and 63xBC or two templates of 9x46PC-36BC. By using 9x46PC-36BC, we could get the same number of postcards and $36 * 2 - 63 = 9$ extra business cards for the same amount of paper. It is not surprising that 9x46PC-36BC is more paper-efficient.

Press sheet templates are also associated with a particular size of paper (e.g. 28"x20"). Therefore, some printing presses will be unable to print press sheets of certain templates because those templates are too large or too small. The set of valid machines for a press sheet template is a subset of the set of all machines: $M_p \subseteq M$

C. UV Coating

Products may also be ordered with spot or full UV coating. This is a glossy coating that is cured using ultraviolet light. We say that each order requires spot, full, or no UV coating. This is for each side of a product, so an order could have full coating on the front and spot coating on the back.

Full coating floods an entire press sheet with UV-sensitive coating, coating all products on the press sheet entirely. Spot coating uses a silkscreen to allow UV coating on particular parts of the press sheet, and even particular parts of an order on that press sheet.

Orders that require spot coating, full coating, or no coating may be placed on a spot-coated press sheet by creating a silkscreen with the area exposed or blocked out where a product requires full or no coating. There is significant time and cost associated with producing a silkscreen, so while we are able to place full UV and no UV orders on a spot press sheet, we would prefer not to if producing a spot press sheet is unnecessary.

D. Attributes

We define the attributes of an order or press sheet to be the combination of the number of order sides and the UV coating options on the front and back sides.

$$a \in A,\ A = \{2\text{SS}, 2\text{SF}, 2\text{SN}, 2\text{FF}, 2\text{FN}, 2\text{NN}, 1\text{S}, 1\text{F}, 1\text{N}\}$$

The first character represents the number of sides. The second and possibly third character determine how the order or press sheet will be UV coated. We use these abbreviations: S for spot coating, F for full coating, and N for no coating. As an example, a 2-sided order with full coating on the front and no coating on the back would have $a = 2\text{FN}$. Another order that requires no coating on the front and full coating on the back would also have $a = 2\text{FN}$, but the product would be flipped on the press sheet.

These attributes apply to both orders and press sheets. Orders may only be placed on press sheets if that press sheet supports it. We say that a press sheet supports an order if the attributes of the order are the same as the attributes of the press sheet. Additionally, two sided press sheets support one sided orders, and spot UV press sheets support full UV or no UV orders. We denote the set of order attributes that are supported by a press sheet with attributes a as $A_a \subseteq A$. For example:

$$A_{2\text{SN}} = \{2\text{SN}, 2\text{FN}, 2\text{NN}, 1\text{S}, 1\text{F}, 1\text{N}\}$$

E. Cutting

After the press sheets have been printed and optionally UV coated, guillotine cutters are used to separate the individual products from each other. Different press sheet templates will require different amounts of time and effort to cut. This contributes to the cost of producing a press sheet. It is possible to create press sheet templates that are highly paper efficient but are very difficult to cut. Effective choosing of press sheet templates will require balancing the cost and time of cutting with the cost of paper.

F. Press Sheets

We define a press sheet by a 4-tuple including a template p, attributes a, press m, and quantity q: (p, a, m, q). These press sheets have real orders placed in the slots that were available on its template t. It will be printed on one or two sides and may have UV coating applied according to a. The press sheet will be printed using the printing press given by m. The image on a press sheet is printed onto individual paper sheets q times.

G. Cost of Press Sheets

To minimize the total cost of production, we need to have estimated costs associated with producing a given press sheet. We denote these costs as $c_{pamq} \in \mathbb{R}$. This is the cost of producing press sheet template p with attributes a on machine m at quantity q.

A number of production tasks and materials are included in the cost of a press sheet:

1) Paper sheets are typically the largest cost.
2) Offset presses require plates to be made. Additionally, offset presses need extra paper, ink, and labor for make-ready.
3) Digital presses are charged by sheet and typically have a higher per-sheet cost than offset presses.
4) Press time and labor costs scale with the quantity of the run.
5) UV coating material is required for both full and spot runs. Silkscreens are required for UV spot runs.
6) Cutting time and labor costs are determined by the template and the quantity of the press sheet.

H. Incoming Orders

During each day, a factory receives new orders that are due at some date in the future. We split the orders into required orders and optional orders depending on if an order is due on the current day. Instead of considering each order separately, we group and count orders by product type, attributes, and quantity. These are the three qualities that allow us to combine orders together onto a press sheet. We denote the number of required orders of product type t and attributes a at quantity q as $v_{taq} \in \mathbb{N}$. We also denote the number of optional orders of product type t and attributes a at quantity q as $w_{taq} \in \mathbb{N}$. We only allow press sheets and orders to be printed at particular discrete quantities.

$$q \in Q,\ Q = \{\dots, 500, 1000, 2000, 2500, \dots\}$$

This decreases the complexity of gang-run printing a set of orders. This is because a properly chosen set Q will increase the chance that orders may be ganged together. If products were able to be ordered at any quantity, it would be much less likely that orders would be able to be ganged together efficiently.

Here we make an important clarification. The values of v_{taq} and w_{taq} are the number of orders at a particular combination of (t, a, q). They are not the number of products ordered. The number of products ordered for each order is given by the q of a (t, a, q) combination. For example, we may have 121 distinct orders of 1000 one-sided business cards with no UV coating that are due on the current day, $v_{\text{'business cards',1N,1000}} = 121$. The total number of business cards that need to be printed is $121 * 1000 = 121000$.

I. Product Placement

An order may be placed one or more times on a press sheet or across multiple press sheets so long as there are slots (u_{pt}) available. If an order is placed n times on a press sheet, we say it is n-up on that press sheet. As an example, if we have an order for business cards at quantity 1000, and a press sheet template that has 63 slots for business cards, we could do one business card run with that order 63-up and print 16 sheets. This would create $16 * 63 = 1008$ business cards in total. In this case, we are producing eight extra business cards. We call this *overprinting*. Additionally we are *splitting* this order by treating a single 1000 quantity order as an equivalent 63 orders of quantity 16.

However, if we have many business card orders at quantity 1000, we would be able to print these orders more efficiently. Instead of making a new press sheet for each order, we could place 63 different orders 1-up on a press sheet and produce 1000 sheets on that run. This would produce 1000 business cards for each order. If we had used the previous method and created a new press sheet for each order, this would require 63 distinct press sheets to produce 63 distinct orders. Here, we use only one press sheet at quantity 1000 to produce the same set of orders. By using only one press sheet, we can save significant costs in plates, setup time, and make-ready.

This way of producing orders is called gang-run printing, because we are combining (ganging) multiple orders onto a press sheet. Gang-run printing has great cost savings because fewer press sheets—and therefore fewer plates, less setup time, and lower make-ready costs—are required to produce a given number of products. The fixed costs of producing a press sheet are distributed among the products that are being produced on that sheet.

While we could print an order at any time after it has been received, to save cost, we try to wait until there are other orders that could be combined with it efficiently before printing. Because of the great cost savings that come from gang-run printing, we try to avoid overprinting and splitting unless we do not have enough orders to fill a press sheet efficiently. Therefore, on any given day we only require printing the orders that are due out that day. We may print orders that are not due if it allows us to print orders more efficiently. The complexity in gang-run printing comes from determining which orders to print, which press sheets to use, and how to split and overprint orders such that printing costs are minimized over time.

J. Calculating an Order's Best Possible Cost

From a gang-run perspective, there is a best press sheet for any given order. Ideally, we would prefer to place every order 1-up on a press sheet that has the same attributes and quantity as the order being placed. We would choose the best press sheet available on the printing press that is cheapest

Constant Name	Description
$t \in T$	Product types
$p \in P$	Press sheet templates
$u_{pt} \in \mathbb{N}$	Number of slots on press sheet template p for product type t
$a \in A$	Attributes
$A_a \subseteq A$	The set of attributes that are supported by a press sheet with attributes a
$m \in M$	Printing presses
$M_p \subseteq M$	Possible presses for press sheet template p
$q \in Q$	Allowed order quantities (discrete set)
$v_{taq} \in \mathbb{N}$	Number of required orders of product type t and attributes a at quantity q
$w_{taq} \in \mathbb{N}$	Number of optional orders of product type t and attributes a at quantity q
$c_{pamq} \in \mathbb{R}$	Cost of producing press sheet template p with attributes a on machine m at quantity q
$k_{taq} \in \mathbb{R}$	Best possible cost of producing an order of product type t and attributes a at quantity q

TABLE I

SUMMARY OF CONSTANTS

Variable Name	Description
$b_{pamq} \in \mathbb{N}$	Number of press sheet template p to produce with attributes a on machine m at quantity q
$r_{taq} \in \mathbb{N}$	Number of orders of product type t and attributes a at quantity q to print
$r'_{taq} \in \mathbb{N}$	Number of orders of product type t and attributes a at quantity q to print (after overprinting)
$r''_{taq} \in \mathbb{N}$	Number of orders of product type t and attributes a at quantity q to print (after splitting)
$r'''_{taq} \in \mathbb{N}$	Number of slots of product type t and attributes a at quantity q to print (after attributes)
$d_{taq_{from}q_{to}} \in \mathbb{N}$	Number of times to treat orders of product type t, attributes a, and quantity q_{from} as an order with quantitity q_{to}
$d'_{taq_{from}q_{to}} \in \mathbb{N}$	Number of times to treat orders of product type t, attributes a, and quantity q_{from} as an order with quantitity q_{to}
$d''_{ta_{from}a_{to}q} \in \mathbb{N}$	Number of times to treat orders of product type t, quantity q, attributes a_{from} as an order with attributes a_{to}

TABLE II

INTEGER DECISION VARIABLES

for the quantity being printed. Remember, digital presses are typically cheaper for lower quantity orders than offset presses. Printing in this manner would best divide the fixed costs of producing a press sheet over the orders being printed on that press sheet.

If all press sheet templates had only one product type, the cost of any individual order on a press sheet could be calculated by simply dividing the cost of the press sheet by the number of slots that are available for products. In this case, the answer would be:

$$k_{taq} = \min_{m \in M,\ p \in M_p} \frac{c_{pamq}}{u_{pt}}$$

This minimizes over the set of machines and templates for the combination that gives the best possible per-product cost.

However, since our templates may have multiple product types, we need some other way to divide the cost of producing a press sheet to each product. We choose to divide this cost according to the fraction of the used paper area. Here, z_t is the area in square inches used by a product type.

$$k_{taq} = \min_{m \in M,\ p \in M_p} \frac{c_{pamq} z_t}{\sum_{t' \in T} u_{pt'} z_{t'}}$$

If the press sheet template has only one product type, this is equivalent to the previous formula, since, in this case, $\sum_{t' \in T} u_{pt'} z_{t'} = u_{pt} z_t$.

The calculated values of k_{taq} provide a lower bound on the cost of producing a particular product. If there were an infinite number of all kinds of incoming orders, we could minimize the total cost of production by choosing the templates and machines that would minimize k_{taq} for each order type. In the next section, we present an integer linear programming formulation that minimizes cost when there are not an infinite number of orders.

III. PROBLEM FORMULATION

A. Problem Statement

We consider printing over multiple days. Each day, we receive new incoming orders that must be printed by some day in the future. We require that the orders that are currently due must be printed during the current day. The orders that are not currently due may be printed, but it is not required. For each day, we must also choose a set of press sheets to print the orders we have decided to print for that day.

Instead of simply printing only the orders that are due, additionally printing orders that are due in the future may allow us to more efficiently combine orders onto press sheets. For example, if only one postcard order of 500 quantity is due for the current day, it would generally be cost-efficient to also print other postcards that are not due for the current day to fill in remaining slots on any press sheets that we choose to print.

B. Solution Description

This problem is solved in two parts. First, the problem is modeled as an Integer Linear Program (ILP). We have a linear objective function, linear constraints, and integer constraints on decision variables. The ILP decision variables will determine how many products to print, which press sheets to build, and how to split and combine orders onto those press sheets. The decision variables do not say which orders should be placed on which press sheets, only how many press sheets should be made of each (p, a, m, q) combination for that day. Similarly, they do not say which particular orders should be printed, only how many orders of each (t, a, q) combination should be printed. These integer-valued variables are described in Table II.

After a suboptimal integer solution to the ILP problem is found, it is used as input to a post-processing stage. This post-processing stage is described in subsection III-D.

C. ILP Problem

The problem is modeled using the integer linear program given here:

$$\min_{\substack{b,d,d',d'' \\ r,r',r'',r'''}} \sum_{p\in P}\sum_{a\in A}\sum_{m\in M_p}\sum_{q\in Q} c_{pamq}b_{pamq} \tag{1a}$$

$$+\sum_{t\in T}\sum_{a\in A}\sum_{q\in Q} k_{taq}(v_{taq}+w_{taq}-r_{taq})$$

$$\text{subject to } v_{taq} \leq r_{taq} \leq v_{taq}+w_{taq} \tag{1b}$$

$$r_{taq} = \sum_{\substack{q_{to}\in Q \\ q\leq q_{to}}} d_{taqq_{to}} \tag{1c}$$

$$r'_{taq} = \sum_{\substack{q_{from}\in Q \\ q_{from}\leq q}} d_{taq_{from}q} \tag{1d}$$

$$r''_{taq} = r'_{taq} - \sum_{\substack{q_{to}\in Q \\ (q-qto)\in Q \\ q/2\leq q_{to}<q}} d'_{taqq_{to}} + \sum_{\substack{q_{from}\in Q \\ (q-q_{from})\in Q \\ q_{from}/2\leq q\leq q_{from}}} \left(d'_{taq_{from}q} + d'_{taq_{from}(q-q_{from})}\right) \tag{1e}$$

$$r''_{taq} = \sum_{a\in A_{a_{to}}} d''_{taa_{to}q} \tag{1f}$$

$$r'''_{taq} = \sum_{a_{from}\in A_a} d''_{ta_{from}aq} \tag{1g}$$

$$r'''_{taq} \leq \sum_{p\in P}\sum_{m\in M_p} u_{pt}b_{pamq} \tag{1h}$$

The objective function (1a) is the sum of two components. The first component is the cost of producing the press sheets that are to be made for the current day. This is a sum of the cost of a press sheet (c_{pamq}) times the number of press sheets that are being made (b_{pamq}) across all possible press sheet types (p, a, m, q).

The second component is a lower bound on the cost of producing the orders that are not being printed on this current day. We multiply the number of orders that are not being printed ($v_{taq} + w_{taq} - r_{taq}$) by a lower bound on producing those orders (k_{taq}) for each possible order combination (t, a, q) and sum the result.

Again, the goal is to minimize the total cost of production over multiple days. As a simplification, we minimize the total cost of production for the current day plus a lower bound on the cost of all optional items not being printed. This ensures that optional items will be included if they do not increase the current cost of production. For example, if there is an empty slot on a press sheet, it would not increase our current production costs to pull in an optional item, but it would decrease our overall cost as measured by this objective function.

The first constraint (1b) ensures that we are printing at least the required orders and at most all available orders. Optional orders that we do not produce during this day must be produced at some time in the future. The last constraint (1h) ensures that we have enough slots on the press sheets we are creating for the orders we have chosen to print. We sum the number of slots of each press sheet template times the number of press sheets that are being created from that template, across all presses. As long as there are at least as many slots available as there are orders to be printed, this constraint is satisfied.

The remaining constraints allow for order overprinting, splitting, and placing orders on press sheets that support its attributes. These constraints relate r_{taq} with r'''_{taq}. The r_{taq} and r'''_{taq} variables are in some way equivalent. The r_{taq} variables describe how many orders of each type will be printed, while the r'''_{taq} variables describe how those orders are being treated for printing. For example one 1000 quantity 1N business card order ($r_{\text{'business cards',1N,1000}} = 1$) may be treated as two 500 quantity 1S business card orders ($r'''_{\text{'business cards',1S,500}} = 2$). If $r_{taq} = r'''_{taq}$ for each (t, a, q) combination, each order would be placed 1-up on a press sheet with the same quantity and attributes as the order. These constraints are described in greater detail in the next subsections.

1) Overprinting: The two constraints 1c and 1d allow orders to be "overprinted." This means they could be placed 1-up on a press sheet of quantity higher than the order requires. This should generally be avoided, but there may be cases where this would decrease overall cost.

Remember, r is the number of orders that will be printed. The variable r' describes those same orders, but with some orders moved to (t, a, q) combinations of higher quantity. The $d_{taqq_{to}}$ variables determine how many orders of (t, a, q) will be printed at quantity q_{to}. The first constraint (1c) ensures that all of the orders to be printed are accounted for by the d variables. The second constraint (1d) collects the d variables into the r' variables.

For instance, a 2000 quantity order may be placed on a 2500 quantity press sheet, with the extra 500 items being discarded, leading to the following assignment of variables:

$$\begin{aligned} &r_{ta,2000} = 1 \quad d_{ta,2000,2500} = 1 \qquad r'_{ta,2000} = 0 \\ &r_{ta,2500} = 0 \qquad\qquad\qquad\qquad\quad r'_{ta,2500} = 1 \end{aligned}$$

2) Split Orders: The constraint 1e allows orders to be split down to press sheets of lower quantity. Orders that are split become equivalent to two smaller orders with quantities that sum up to the original. This constraint is a bit different from the constraints that allow overprinting and printing on press sheets of different attributes. The first sum takes orders away from a given (t, a, q) combination. Orders from other (t, a, q) combinations are collected in the second sum, which not only takes orders from q_{from} but also $q - q_{from}$. This method of splitting will only work well if the set Q contains differences between two quantities in Q. Placing both of these sums together in the single constraint that relates r' and r'' allows splitting to happen multiple times. For instance, a 2500 quantity order may be split to a 1500 quantity order

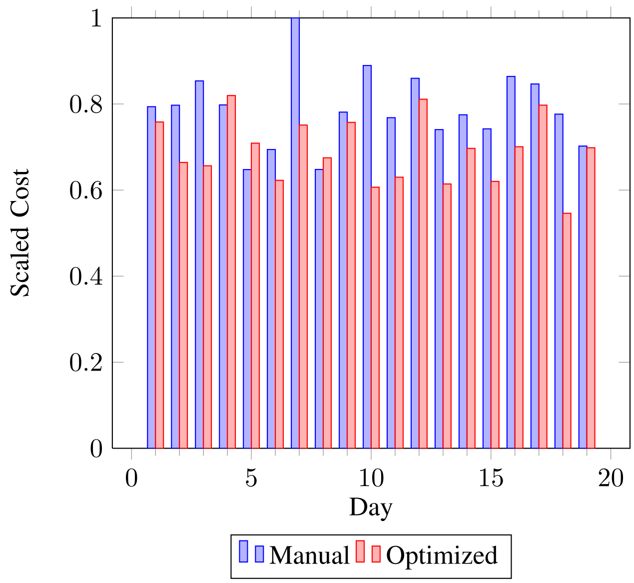

Fig. 2. Total cost comparison between manual and optimized press sheet selection, scaled by the highest cost day. For each day, the first bar represents the manual cost and the second bar represents the optimized cost.

and two 500 quantity orders:

$$\begin{array}{lll} r'_{ta,2500} = 1 & & r''_{ta,2500} = 0 \\ r'_{ta,1500} = 0 & d'_{ta,2500,1500} = 1 & r''_{ta,1500} = 1 \\ r'_{ta,1000} = 0 & d'_{ta,1000,500} = 1 & r''_{ta,1000} = 0 \\ r'_{ta,500} = 0 & & r''_{ta,500} = 2 \end{array}$$

3) Attributes: The constraints 1f and 1g allow orders to be printed on press sheets with attributes that support them. They function similarly to the constraints for overprinting. For instance, a 1F order could be placed on a 2FN press sheet:

$$\begin{array}{lll} r''_{t,\text{1F},q} = 1 & d''_{t,\text{1F,2FN},q} = 1 & r'''_{t,\text{1F},q} = 0 \\ r''_{t,\text{2FN},q} = 0 & & r'''_{t,\text{2FN},q} = 1 \end{array}$$

D. Post-processing

After a sufficiently good solution to the problem has been found by an ILP solver, we still need to actually place the orders onto real press sheets. We create the correct number of press sheets of the right type according to b_{pamq}. All required orders are used along with some optional orders according to r_{taq}. We follow the constraints that transform r_{taq} into r'''_{taq} by using the d, d', and d'' decision variables to determine how many orders from a (t, a, q) combination should be grouped with other orders of another (t, a, q) combination. After all orders have been moved or split to their final (t, a, q) combination, we place the orders onto press sheets that support them. The constraints from the ILP problem ensure that we have enough slots available on the press sheets to place all orders onto press sheets.

IV. RESULTS / CONCLUSIONS

This method was compared using four weeks of historical order data from a real factory. The factory has a number of digital and offset presses of varying sizes. Press sheet costs were calculated using real paper, plate, ink, UV coating/screen, and labor costs.

We used Gurobi[11] to find feasible solutions and allowed Gurobi to optimize for five minutes per day. This was run on an Intel i7 3770K processor with 16GB of RAM using Gurobi version 5.0.0. with the default Gurobi options. Except for very small instances of the problem, Gurobi was not able to prove optimality within a reasonable amount of time. For the days listed above, the optimality gap was typically between 0.1% and 1%. The solutions produced by Gurobi were compared against the real press sheets that were selected by hand during those weeks.

Fig. 2 shows the per-day cost comparison between the two methods. This figure is scaled to show relative differences between the two methods but not disclose the total order volume for the factory. Our method is estimated to save a total of 14% of real dollars in cost over these weeks. Most of the cost savings came from selection of more paper-efficient press sheet templates. Our method also did much better at minimizing the number of needed UV screens for spot runs by better combining orders with full or no coating.

REFERENCES

[1] N. Bansal, L. A., and M. Sviridenko, "A tale of two dimensional bin packing," in *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, oct. 2005, pp. 657–666.
[2] H. Hindi and L. Crawford, "Control in printing systems: Modular reconfigurable media paths," in *American Control Conference, 2009. ACC '09*, june 2009, pp. 38–52.
[3] K. Nolde and M. Morari, "Modeling and control of thermal printing," *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 2, pp. 405–413, march 2010.
[4] M. Agrawal, Q. Duan, K. Chakrabarty, J. Zeng, I.-J. Lin, G. Dispoto, and Y.-S. Lee, "Digital print workflow optimization under due-dates, opportunity cost and resource constraints," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, july 2011, pp. 86–92.
[5] S. Rai, C. B. Duke, V. Lowe, C. Quan-Trotter, and T. Scheermesser, "LDP lean document production—O.R.-enhanced productivity improvements for the printing industry," *Interfaces*, vol. 39, no. 1, pp. 69–90, January/February 2009.
[6] R. Zhou, S. Rai, and M. Do, "Scheduling print jobs in lean document production (LDP) toolkit," presented at the 21st International Conference on Automated Planning and Scheduling (ICAPS-11), Freiburg, Germany, June 2011.
[7] J. Zen, I.-J. Lin, E. Hoarau, and G. Dispoto, "Numerical simulation and analysis of commercial print production systems," in *25th International Conference on Digital Printing Technologies (NIP25)*, vol. 25, sept. 2009, pp. 581–584.
[8] I. Lithotechnics, "Metrix," 2012. [Online]. Available: http://metrixsoftware.com
[9] H. Kipphan, *Handbook of Print Media: Technologies and Production Methods*. Springer, 2001.
[10] R. Chung and M. Rees, *A Survey of Digital and Offset Print Quality Issues*, ser. Research monograph. Printing Industry Center, RIT, 2007.
[11] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2012. [Online]. Available: http://www.gurobi.com