Lot streaming in a multiple product permutation flow shop with intermingling

Dirk Biskup, Martin Feldmann

Lot streaming in a multiple product permutation flow shop with intermingling

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2006-IJPR-0474.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	15-Jul-2006
Complete List of Authors:	Biskup, Dirk; Bielefeld University, Department of Business Administration and Economics Feldmann, Martin; Bielefeld University, Department of Business Administration and Economics
Keywords:	PRODUCTION PLANNING, INTEGER PROGRAMMING, LOT STREAMING, SCHEDULING
Keywords (user):	

Lot streaming in a multiple product permutation flow shop with intermingling

Martin Feldmann

Dirk Biskup*

Department of Business Administration and Economics
Bielefeld University
Postfach 10 01 31
33501 Bielefeld, Germany
dbiskup@wiwi.uni-bielefeld.de

Abstract

In this paper we study the multi-product lot streaming problem in a permutation flow shop. The problem involves splitting given order quantities of different products into sublots and determining their optimal sequence. Each sublot has to be processed successively on all machines. The sublots of the particular products are allowed to intermingle, that is sublots of different jobs may be interleaved. A mixed integer programming formulation is presented which enables us to find optimal sublot sizes as well as the optimal sequence simultaneously. With this formulation small and medium sized instances can be solved in a reasonable time. The model is further extended to deal with different settings and objectives. As no lot streaming instances are available in the literature, LSGen, a problem generator is presented, facilitating valid and reproducible instances. First results about average benefit of lot streaming with multiple products are presented, which are based on a computational study with 160 small and medium sized instances.

Keywords: lot streaming; scheduling; production planning; mixed integer programming

*Corresponding author

1. Introduction and literature review

The term "*lot streaming*" denotes techniques of splitting given jobs, each consisting of identical items, into sublots to allow their overlapping processing on successive machines in a multi-stage production system. While traditional scheduling problems assume that jobs or lotsizes are fixed, lot streaming problems can be considered as sequencing problems with the characteristic that the magnitude of each sublot is a decision variable. In line with Allahverdi et al. (1999), these techniques are part of job floor control, where the master production schedule has to be realized. Lot or batch sizes are specified by the production planning and control system, but regularly these targets turn out to be infeasible during execution. One option to dealing with this problem is the application of lot streaming procedures, i.e. items are rearranged and allocated in sublots. If these sublots are produced in an overlapping fashion, remarkable reduction of makespan and improved timeliness are within reach (Kalir/Sarin, 2000). Due to its high relevance, Lee et al. (1997) classify lot streaming as one of the current trends in deterministic scheduling. They point out the necessity to extend classical algorithms to models which are more closely related to real world problems.

The first formal results on lot streaming are obtained by dealing with the one-product-case in a flow shop with two and three stages (Potts/Baker, 1989). In the concluding part of their paper Potts/Baker address the problem of lot streaming with two products on two stages. They give a small example to show that sequential decisions –first sequencing the jobs without lot streaming and afterward applying lot streaming individually to each job– may lead to suboptimal schedules. However, Potts/Baker (1989) did not present a general solution procedure for streaming with multiple products. The vast majority of research in lot streaming has been concerned with the one-product-case only. A comprehensive and excellent review of well solved variants in lot streaming is given by Trietsch/Baker (1993) – for more recent literature reviews see Biskup/Feldmann (2005), Chang/Chiu (2005) and Feldmann (2005).

Generally, the goal in lot streaming is to determine the number of sublots for each product, the size of each sublot and the sequence for processing the sublots so that a given objective is optimized (Zhang et al., 2005). As the general problem remains unsolved, research typically tackles less general versions of the general lot streaming problem. The following terms summarize different directions of lot streaming research, see Potts/Van Wassenhove (1992), Trietsch/Baker (1993), Kalir/Sarin (2001) and Zhang et al. (2005):

- **Single product / multiple products:** Either a single product or multiple products are considered.
- **Fixed / equal / consistent / variable sublots:** *Fixed sublots* means that all sublots for all products consist of the identical number of items on all stages. *Equal sublots* means that sublot sizes are fixed for each product. The differentiation between fixed and equal sublots is only necessary for multiple products. A sublot is called *consistent* if it does not alter its size over the stages of processing. For *variable sublots* no restrictions are given.
- **Non-idling / intermitted idling:** For *non-idling* the sublots on a particular stage have to be processed directly one after the other. For *intermitted idling* on the other hand, idle times between sublots may occur.
- **No-wait / wait schedules:** In *no-wait schedules*, each sublot has to be transferred to and processed on the next stage immediately after it has been finished on the preceding stage. In a *wait schedule*, a sublot may wait for processing between consecutive stages.
- **Attached setups / detached setups / no setups:** If *attached setups* are required the setup can not start until the sublot is available at the particular stage. In a *detached setup* the setup is independent from the availability of the sublot. And sometimes setup times are neglected or do not occur.
- **Discrete / continuous sublots:** For *discrete sublots*, the number of items of a sublot has to be an integer. For *continuous sublots* no such restriction exists.
- **Intermingling / non-intermingling sublots**: If in a multi-product setting *inter-mingling sublots* are allowed, the sequence of sublots of product *j* may be interrupted by sublots of produkt *k*. For *non-intermingling sublots* no interruption in the sequence of sublots of a product is allowed, which is obviously always given in one-product settings and can be forced in multi-product settings.

In the following, we survey research on **multi-product** lot streaming problems and focus on flow shop environments, and consider consistent or variable sublots results in a magnitude of related problems:

Vickson/Alfredsson (1992) consider multiple products on two and three stages with unit-sized sublots, i.e. every item has to be transferred separately. If setup and transfer times are negligible and regular measures of performance are used, unit-sized sublots are proved to be optimal. Moreover, but restricted to the two-stage setting, it can be shown that optimal schedules exist with non-intermingling sublots, but if the number of stages increases, optimal solutions may require intermingling sublots (Vickson/Alfredsson, 1992, p. 1564). Vickson (1995) considers non-intermingling sublots on two stages and investigates the question of how to solve lot streaming problems with job or sublot detached setups and attached setups for discrete and consistent sublots, respectively. He presents some closed form solutions for continuous sublots and a fast polynominally bounded search algorithm for discrete sublots. Baker (1995) continues the analytic work of Vickson/Alfredsson (1992) by incorporating sublot-attached setup times into the model. He exploits some theoretical results of scheduling with time lags, but his findings strongly rely on the fact that in two-stage settings, permutation schedules are known to be optimal. For more than two stages, optimality is no longer guaranteed.

Lot streaming with multiple products and fixed sublot sizes is intensively discussed by Kalir (1999). In the case of continuous and fixed sublots, closed forms can be given for the optimal number of sublots and sublots-sizes, respectively. Kalir/Sarin (2001) present the BMI heuristic to sequence fixed sublots in multi-stage flow shops, if sublots are not allowed to intermingle. This heuristic constructs a schedule which attempts to minimize idle time on the bottleneck machine. Kalir/Sarin (2003) deal with sublot-attached setups, while equal and non-intermingling sublots are assumed. They present a solution procedure which finds optimal solutions if one product is streamed on two stages. They further propose procedures to gain near optimal solutions with equal, non-intermingling sublots for multiple products on two stages by applying Johnson's rule (Johnson, 1954). Moreover, they discuss an extension of their approach to the multi stage setting, modifying the BMI heuristic.

Lee et al. (1993) minimize makespan in a multi-stage lotsizing and scheduling problem with significant and sequence depending setup times. The total lot size of each product is assumed to be given and items are allowed to be produced in an overlapping fashion – so their problem is equivalent to lot streaming with consistent and intermingling sublots in a permutation flow shop. They develop a genetic algorithm and focus their research on the effect of an evolving chromosome structure, where building blocks are directly interpreted as lot-sizes: In the beginning, a randomly generated sequence of fix and minimal lot sizes (e.g. 5 items per sublot) for all products is given. During the search, positions of sublots are interchanged and consecutive sublots of the same product are aggregated if and only if this aggregation is advantageous. As re-splitting of aggregated sublots is not modelled, sublot sizes are only allowed to increase. Nevertheless, sequencing and lot sizing are decided simultaneously, sublots are allowed to intermingle, and finally the number of sublots for every product is adjusted by the genetic algorithm. However, sublot sizes are restricted to be multiples of the given minimal fixed sublot size, and the approach does not guarantee to find optimal solutions.

Kumar et al. (2000) consider the multi-product, multi-stage, no-wait flow shop with non-intermingling discrete sublots. Their solution procedure consists of three-steps: First, optimal, consistent and continuous sublots are calculated separately for every product by linear programming. Secondly the sublots are rounded, as discrete sublots are required. In the third step the remaining sequencing problem among the products is reformulated as a TSP and solved heuristically. The approach of Kumar et al. (2000) generalizes the procedure of Sriskandarajah/Wagneur (1999), which is restricted to two-stage settings, detached setups and consistent (continuous as well as discrete) lot sizes. In addition, Kumar et al. (2000) present two genetic algorithms to solve the sublot size or the product sequencing task. They further develop some hybrid heuristic approaches (combinations of genetic search, linear programming and heuristical TSP procedures) and allow the number of sublots to be adjusted during the search. Hall et al. (2003) study the problem of Sriskandarajah/Wagneur (1999) with attached setups and develop an efficient heuristic to solve the multi-stage no-wait lot streaming problem with multiple products, if consistent non-intermingling but integer sublot sizes are assumed.

Only few studies on production environments other than flow shops are available:

- Zhang et al. (2005) deal with lot streaming in *m-1 hybrid flow shops* to minimize mean completion time. On the first stage m identical and parallel machines are given, while the following stages are arranged like a traditional flow shop. In their study only two stages are investigated: two parallel machines are given on the first stage and one machine on the second stage. Each sublot requires a setup. Similar to the paper of Kumar et al. (2000), the number of sublots is a decision variable and sublot sizes are restricted to be larger than a fixed minimal sublot size. They present two heuristic approaches and a MIP model, but again sublots are not allowed to intermingle.
- Lot streaming in *job shop* environments is dealt with by Dauzère-Pérès/Lasserre (1997). They propose an iterative procedure, where first lot streaming with consistent sublots is executed, and in a second step the scheduling decisions are regarded. As job shop

scheduling is NP-hard, Dauzère-Pérès/Lasserre apply the shifting bottleneck heuristic (Adams et al., 1988).

- Lot streaming in *open shops* was first considered by Şen/Benli (1999). They present some results for scheduling a single job in multi-stage open shops, considering single or multiple routing for each sublot. Furthermore they focus on the multiple-job lot streaming problem with two stages and show that lot streaming will only improve makespan if there is a job with large processing times. Close form solutions are given to calculate optimal sublot sizes and their sequences. Hall et al. (2005) study the problem of minimizing makespan in no-wait two-machine open shops with consistent and non-intermingling sublots by modifying the procedures given in Hall et al. (2003). As the problem additionally requires a machine sequence for each product, the study is restricted to two stage settings. A dynamic programming algorithm is used to generate all dominant schedule profiles for each product. These profiles are required to formulate the open shop problem as a generalized traveling salesman problem. A computationally efficient heuristic is presented and it is shown that good solutions can quickly be found for two machine open shops with up to 50 products.

Recapitulating the solution status of lot streaming problems, one important aspect – already highlighted by Potts/Baker (1989) – is still open. It is the question of how to find optimal solutions in a multi-stage multi-product flow shop if sublots are allowed to intermingle. In line with the studies mentioned above, we consider a permutation flow shop to let the sequencing decision only occur once, and restrict sublot sizes to consistent sublots. From a practical point of view permutation flow shops have two big advantages: Firstly, as the sequencing decision is determined on the first stage, the remaining stages do not need to bother with (error-prone) sequencing issues. Their schedule is given by the order the different sublots arrive. Secondly, from the perspective of fast error detection sublots need to be easily tracable. Obviously lot tracing is much easier in permutation flow shops than in open flow shops (Feldmann, 2005, p. 71).

In contrast to the studies mentioned above, our mixed integer programming formulation simultaneously determines the lot sizes and the sequence of sublots to guarantee overall optimal solutions. To the best of our knowledge the complexity status of the lot streaming problem considered in this paper is still open - but as makespan minimization in permutations flow shop scheduling is known to be NP-hard for three and more machines (Garey, et al.

1976), the problem under study is most probably NP hard (Trietsch/Baker, 1993, Sriskandarajah/Wagneur, 1999), too.

The remainder of the paper is organized as follows: In the next section we introduce a model formulation for the multi-stage multi-product flow shop problem with sublots that are allowed to intermingle. This model formulation is afterwards extended to some settings that seem to be very interesting from a practical point of view. In the third section we discuss the benefits of lot streaming by introducing a problem generator and solving 1,760 problems to optimality. The paper concludes with some final remarks in section four.

2. Model Formulation and Extensions

With the following model formulation, generally speaking, the two inherent goals of the problem, namely determining the sequence among the sublots and the size of the individual sublots, are solved simultaneously. We will make use of the following variables and symbols:

S := number of sublots per product

s,t := indices for the sublots, $s, t = 1, \ldots, S$

M := number of machines

m := index for the machines, $m = 1, \ldots, M$

J := number of products

j,k := indices for the products, $j, k = 1, \ldots, J$

r_{jm} := processing time for one unit of product j on machine m

u_{js} := number of units produced in sublot s of product j

p_{jsm} := processing time of sublot s of product j on machine m

L_j := number of identical items of product j to be produced

R := sufficiently large number

b_{jsm} := starting time of the sublot s of product j on machine m

x_{jskt} := binary variable, which takes the value 1 if sublot s of product j is sequenced prior to sublot t of product k, 0 otherwise

Note that the use of sequence-related binary variables bears some similarity to Manne's (1960) formulation of the job shop scheduling problem. The multi-stage multi-product flow shop problem with sublots that are allowed to intermingle can now be formulated:

Minimize Z

subject to

(1)	$\sum_{s=1}^{S} u_{js} = L_j$	$j = 1, ..., J$
(2)	$p_{jsm} = u_{js}\, r_{jm}$	$j = 1, ..., J;\ s = 1, ..., S;\ m = 1, ..., M$
(3.1)	$b_{jsm} + p_{jsm} \le b_{ktm} + (1 - x_{jskt})R$	$j, k = 1, ..., J;\ j < k;\ s, t = 1, ..., S;\ m = 1, ..., M$
(3.2)	$b_{ktm} + p_{ktm} \le b_{jsm} + x_{jskt}R$	$j, k = 1, ..., J;\ j < k;\ s, t = 1, ..., S;\ m = 1, ..., M$
(4)	$b_{jsm} \ge b_{js,m-1} + p_{js,m-1}$	$j = 1, ..., J;\ s = 1, ..., S;\ m = 2, ..., M$
(5)	$b_{jsm} \ge b_{js-1,m} + p_{js-1,m}$	$j = 1, ..., J;\ s = 2, ..., S;\ m = 1, ..., M$
(6)	$x_{jsk,t-1} \le x_{jskt}$	$j, k = 1, ..., J,\ j < k;\ s = 1, ..., S;\ t = 1, ..., S\text{-}1$
(7)	$Z \ge b_{jSM} + p_{jSM}$	$j = 1, ..., J$
(8)	$x_{jskt} \in \{0,1\}$	$j, k = 1, ..., J,\ j < k;\ s, t = 1, ..., S$
(9)	$u_{js}, b_{jsm} \ge 0$	$j = 1, ... J,\ s = 1, ..., S,\ m = 1, ..., M$

Restrictions (1) ensure that in sum L_j items are processed of product j. With (2) the processing times of the sublots are calculated. Restrictions (3.1) and (3.2) determine the sequence of sublots. Since it is a permutation flow shop, no machine index is needed for x. (3.1) is binding if (and only if) x_{jskt} takes the value 1. In this case sublot s of product j is scheduled prior to sublot t of product k on machine m and the processing of sublot t of product k is forced to start after sublot s of product j has been finished. If, on the other hand, x_{jskt} takes the value zero, (3.1) are not binding, as R is added on the right hand side. The disjunctive counterpart is reflected by restrictions (3.2). These restrictions are only binding, if x_{jskt} takes the value 0. The restrictions (4) and (5) assure that the sublots of the same product do not overlap: With restrictions (4) sublot s on machine m is not allowed to start before sublot s on machine m - 1 has been finished. Restrictions (5) prevent that two sublots, s and $s - 1$, are processed simultaneously on one machine. From a computational point of view, is it advantageous to decrease the number of possible permutations of the binary variables. As stated in (6), an inherent structure among the variables x_{jskt} is known: If sublot s of product j is scheduled prior to sublot t of product k, sublot s must also be scheduled prior to sublot t + 1, t + 2, ..., S of product k. With the restrictions (6) the number of iterations (LINGO 7.0 is used) could be reduced to approx. 60% compared to the model without them. In (7) the completion time of the last sublot *S* on the last machine M are used to define the makespan Z.

In line with most of the literature on lot streaming we assume that sublots do not need to be discrete, see (9). However, discrete sublots can easily be generated by non negative integer requirements for u_{js}, $j = 1, \ldots, J$, $s = 1, \ldots, S$ in (9). From a practical point of view there are examples for both cases: Books, cars, furniture, etc. require integer variables while for the production (not the sizing) of gas, beverages, concrete, electricity etc. real variables are appropriate.

The number of binary variables needed can be calculated by $\frac{S^2 \cdot J \cdot (J-1)}{2}$. Note that the number of machines does no impact this formula as a permutation work flow is considered. But with an increasing number of products and sublots the number of binary variables needed increases rather fast, see Table 1:

J\S	2	3	4	5	6	7	…	10
2	4	9	16	25	36	49		100
3	12	27	48	75	108	197		300
…								
10	180	405	720	1,125	1,620	2,205		4,500

Table 1: Number of binary variables needed depending on J and S.

From the perspective of intermingling especially the following settings seem to be of interest:

Extension #1: No intermingling between the sublots of one (or more) of the J products with the other products

All sublots of one (or more) of the *J* products are produced one after the other and are not allowed to intermingle with the other products. This setting might be advantageous if the set-up costs for one or more products are high. Let us assume product three is not allowed to intermingle (and $J = 3$, $S = 3$). A possible sequence on the machines might be: (1_1, 1_2, 2_1, 1_3, 3_1, 3_2, 3_3, 2_2, 2_3). The first number indicates the product, the second number the sublot. To formulate a situation like this we can use the restrictions (3.1) and (3.2) of the above model formulation for all products j and k that are allowed to intermingle. We assume that J_i contains all products that are allowed to intermingle and the subset J_n contains the products that are not allowed to intermingle, i.e. $J = \{J_i, J_n\}$:

(3.1) $b_{jsm} + p_{jsm} \le b_{ktm} + (1 - x_{jskt})R$

$j, k \in J_i;\ j < k;\ s, t = 1, \ldots, S;\ m = 1, \ldots, M$

(3.2) $b_{ktm} + p_{ktm} \le b_{jsm} + x_{jskt}R$

$j, k \in J_i;\ j < k\ \ s, t = 1, \ldots, S;\ m = 1, \ldots, M$

For the products $l \in J_n$ we make use of the following binary variables:

x_{jsl} := binary variable, which takes the value 1 if sublot s of product $j \in J_i$ is sequenced prior to product $l \in J_n$, 0 otherwise

(3.3) $b_{jsm} + p_{jsm} \le b_{l1m} + (1 - x_{jsl})R$

$j \in J_i;\ l \in J_n;\ s = 1, \ldots, S;\ m = 1, \ldots, M$

(3.4) $b_{lSm} + p_{lSm} \le b_{jsm} + x_{jsl}R$

$j \in J_i;\ l \in J_n;\ s = 1, \ldots, S;\ m = 1, \ldots, M$

Furthermore, the definition of the binary variables in (9) has to be adjusted. All other restrictions of the above model formulation apply for both intermingling and non-intermingling products. Another "quick and dirty" approach for this setting was to use the model formulation (1) to (9) and equate the binary variables for the sublots of the product(s) that is (are) not allowed to intermingle. For the above example this would be $x_{js31} = x_{js32} = x_{js33}$, $j = 1, 2$ and $s = 1, 2, 3$.

A model without any intermingling would only make use of the restrictions (3.3) and (3.4). In this case the sequencing part of the problem reduces to finding a sequence among the products (instead of among the sublots).

Extension #2: Overall number of sublots given, but not the number of sublots per product

From a practical point of view a second interesting setting is the following: The overall number of sublots is given but not the number of sublots per product. For example it might, from an logistical perspective, be advantageous to have at most 8 sublots (among $J = 3$ products). Now the task is to find the optimal number of sublots per product, the optimal sequence among the sublots, and the optimal size of the sublots. To formulate a setting like this we make use of position related binary variables.

Let:

P	:=	overall number of sublots allowed, $p = 1, \ldots, P$
x_{pj}	:=	binary variable which takes the value 1 if at the p-th position product j is produced, 0 otherwise
u_{pj}	:=	number of units produced of product j in position p
p_{pjm}	:=	processing time of product j in position p on machine m
b_{pm}	:=	starting time of the product in position p on machine m

The model formulation is as follows:

Minimize Z

subject to

(1') $$\sum_{p=1}^{P-J+1} u_{pj} = L_j \qquad j = 1, \ldots, J$$

(2') $$p_{pjm} = u_{pj}\, r_{jm} \qquad p = 1, \ldots, P;\ j = 1, \ldots, J;\ m = 1, \ldots, M$$

(3') $$\sum_{j=1}^{J} x_{pj} = 1 \qquad p = 1, \ldots, P$$

(4') $$p_{pjm} \le x_{pj} R \qquad p = 1, \ldots, P;\ j = 1, \ldots, J;\ m = 1, \ldots, M$$

(5') $$b_{pm} + \sum_{j=1}^{J} p_{pjm} \le b_{p+1,m} \qquad p = 1, \ldots, P - 1;\ j = 1, \ldots, J;\ m = 1, \ldots, M$$

(6') $$b_{pm} + \sum_{j=1}^{J} p_{pjm} \le b_{p,m+1} \qquad p = 1, \ldots, P;\ j = 1, \ldots, J;\ m = 1, \ldots, M - 1$$

(7') $$Z \ge b_{PM} + \sum_{j=1}^{J} p_{PjM} \qquad j = 1, \ldots, J$$

(8') $$x_{pj} \in \{0, 1\} \qquad p = 1, \ldots, P;\ j = 1, \ldots, J$$

(9') $$u_{pj}, b_{pm} \ge 0 \qquad p = 1, \ldots, P;\ j = 1, \ldots J;\ m = 1, \ldots, M$$

The restrictions (1') ensure that for each product L_j units are produced; note that at most $P - J + 1$ sublots are possible for one job, as for each of the other jobs at least one sublots is necessary. The restrictions (3') allow exactly one product being produced at each of the P positions. This of course means that a positive production time may only occur if the particular binary variable takes the value 1 (4'). All other restrictions are obvious and similar to the model formulation (1) to (9).

Sequence-related binary variables versus position-related binary variables for the multi-stage multi-product flow shop problem with sublots that are allowed to intermingle

The multi-stage multi-product flow shop problem with sublots that are allowed to intermingle can easily be formulated with position-related binary variables as well. At first glance the model formulation (1') to (9') seems to be very compact and easy to solve, and the model formulation with sequence-dependent binary variables (1) to (9) looks more complex. However, it turned out to be far easier to solve (1) to (9) than (1') to (9'). To demonstrate this attribute of the two models, the number of sublots used for every product is restricted by (10'), so both models become comparable.

(10') $$\sum_{p=1}^{P} x_{pj} = \frac{P}{J} \qquad j = 1, \ldots, J$$

In Table 2 the number of branch and bound iterations for both models are given. We solved lot streaming instances with 2, 3 and 4 products. The notation (taken from our problem generator introduced in the following section) indicates the number of products, number of stages and number of instance. For example in instance 2_5_1 two products are streamed over five stages, while instance number 1 is investigated.

instance	Sublots per product	Iterations needed by model (1) to (9)	Iterations needed by model (1') to (10')	%
2_5_1	7	190,450	330,637	173.6
2_5_2	7	155,086	233,450	150.5
2_5_3	7	168,308	320,189	190.2
3_5_1	4	3,229,003	3,990,200	123.6
3_5_2	4	2,133,996	4,066,437	190.6
3_5_3	4	3,714,000	4,122,178	111.0
4_4_1	3	8,918,934	14,807,426	166.0
4_4_2	3	17,678,258	28,017,556	158.5
4_4_3	3	5,318,975	14,550,500	273.6

Table 2: Exemplary comparison of the number of branch and bound iterations of the two models

Considering the instances given in Table 2, the model with position-related binary variables (1') to (10') needs on average significantly more iterations than the model with sequence-related binary variables. We decided not to analyze the difference between the two models further, but to present the formulation (1) to (9) for the multi-stage multi-product flow shop problem with sublots that are allowed to intermingle. If only an overall number of sublots is given, sequence-dependent binary variables cannot be applied with reasonable effort. Therefore we decided to present the model formulation (1') to (9') making use of position related-binary variables for this extension. This model formulation furthermore has the advantage that no-wait and no-idling schedules can be required by formulating (5') or (6') as equations, respectively.

3. Benefit of lot streaming and computational experiments

Studies to evaluate the potential benefit of lot streaming are rare. To the best of our knowledge just two papers tackle this issue:

- Baker/Jia (1993) present a comparative study of over 6,000 test-problems to evaluate the effect of lot streaming in a three stage one-product setting, if non-idling is assumed or consistent sublots or equal sublots and non-idling are given. They found diminishing improvements in makespan reduction for an increasing number of sublots. For every solution procedure, more than half of the potential makespan reduction from ten sublots is obtained with just two sublots, while 80% of the benefit of ten sublots is already obtained with three sublots (Baker/Jia, 1993, p. 565).
- Kalir/Sarin (2000) present some approximation forms for the evaluation of the potential consequences, if one or multiple products are streamed in a flow shop. If equal sublot sizes are assumed, it becomes possible to gain upper-bounds for makespan, mean flow time and work-in process in the single product case. Regarding multiple products, the problem is approachable only if an identical, i.e. product-unspecific, bottleneck machine exists and non-intermingling and unit sized sublots are used. Solely for this limited setting approximative upper-bounds on the benefit of lot streaming are derived.

We are not aware of any results on the benefit of lot streaming with multiple products in a multi-stage setting for consistent sublots. Moreover, no reproducible instances exist in the literature. Along with our computational results we decided to develop a problem generator – called LSGen– to make our computational results reproducible. Furthermore the possibility to replicate benchmark instances may serve as a base for future research on "larger" problems. LSGen can easily be downloaded via the following link: http://www.wiwi.uni-bielefeld.de/%7Ekistner/mitarbeiter/feldmann/lsgen.exe[1]. Within LSGen it is just necessary to appoint the number of products *J,* the number of stages *M* and the number of the instance *N*, to receive the reproducible instance *J_M_N*. LSGen calculates r_{jm} and L_j, as uniformly distributed integers within the following ranges: r_{jm} = {1, ..., 12}; L_j = {10, ..., 40}. Additionally, a $J \times J$ matrix with c_{jk} = {0, ..., 30} is given, if sequence dependent setup times are applied. The pseudo-random numbers used in LSGen are initialized with a seed, calculated as a function in *S*, *M, N* to assure that all instances are calculated independent to other instances and that bigger and smaller instances do not systematically share common properties: seed = 3,965,481 + 1,000**J* + 100**N* + *M* . In the following the data of instance 3_4_10 (three jobs: $J = 3$, four stages: $M = 4$, tenth instance: $N = 10$) are given:

$$r_{jm} = \begin{bmatrix} 3 & 10 & 12 & 7 \\ 10 & 1 & 6 & 10 \\ 6 & 3 & 12 & 11 \end{bmatrix} \qquad L_j = \begin{bmatrix} 32 \\ 38 \\ 18 \end{bmatrix} \qquad c_{jk} = \begin{bmatrix} 24 & 17 & 5 \\ 10 & 0 & 17 \\ 3 & 15 & 9 \end{bmatrix}$$

In Figure 1 the Gantt Chart of instance 3_4_10 depicting an optimal solution with four discrete, intermingling and consistent sublots (no setups) is given. The optimal makespan is Z* = 909.

<< please insert Figure 1 here >>

In this solution the four sublots of job 3 are scheduled first. Then the first sublot of job 2 (named 2_1) follows, but job 2 is intermingled by sublots of job 1. The following sequence and sublot-sizes are found to be optimal: $u_{31} = 3$; $u_{32} = 4$; $u_{33} = 6$; $u_{34} = 5$; $u_{21} = 10$; $u_{11} = 7$; $u_{22} = 11$; $u_{12} = 9$; $u_{23} = 4$, $u_{24} = 13$; $u_{13} = 9$; $u_{14} = 7$. The optimal makespan without intermingling sublots is 1,071, which equates to a disadvantage of 17.8 %.

Overall we generated and solved 160 instances (J = {2, 3}; M = {3, 4,... , 10} N = {1, 2,... , 10}). The number of sublots *S* was set to be in the interval {1, 2,... , 7} for *J* = 2 and *S* = {1, 2, ..., 4} for those instances with *J* = 3. Consequently 880 lot-streaming problems were solved. Additionally, all calculations are repeated for the non-intermingle case, so in total 1,760 optimal schedules form the basis for the statistical evaluation. For these settings solutions with and without intermingling can be found within a second and up to 45 minutes applying

[1] We will gladly distribute LSGen or the collection of instances, used in this paper by mail.

LINGO 7.0 on a standard PC (Pentium 4, 1.8 GHz, Windows 2000). In the following we survey average results. The details are given in the Appendix.

First, we investigate whether an increase in S will show a slope that corresponds to the findings given by Baker/Jia (1993) and whether the problem size will show any effect on the benefit of lot streaming. In Figure 2 the averaged **marginal** benefit of additional sublots is shown. The marginal benefit mb_S is calculated by: $mb_S = (Z_S - Z_{S+1}) / Z_S$ where Z_S denotes the optimal makespan for lot streaming with S consistent sublots. Hence, mb_S denotes the percentage reduction of Z_S if one additional sublot (Z_{S+1}) is allowed. All data of Figure 2 are averaged over 10 instances. For example, among the first ten benchmark problems with $J = 2$ and 6 stages, i.e. 2_6_1, 2_6_2, …, 2_6_10, allowing two sublots, reduces the makespan by 34.69% compared to the situation without sublots (i.e. one production lot). Allowing three sublots reduces the makespan by an additional 17.21% compared to the situation with two sublots.

<< please insert Figure 2 here >>

The benefit of lot streaming in multi-stage settings increases not only with the number of sublots but also with a growing number of stages, see Figure 2. This pattern holds across all numbers of sublots, i.e. the effect of the 4th additional sublot in an eight stage setting is on average higher than the effect of the 4th sublot in a three stage setting. This finding gives important advice to production managers if they have to decide which of the production lines should be accelerated by lot streaming. Considering 10 stage settings, streaming of two products in two sublots reduces makespan compared to the situation without lot streaming by 39% on average while in three stage settings an improvement of only 25% can be realized. The results for lot streaming with three products show the same pattern, thus we decided to omit them.

The averaged **total** benefit of lot streaming is given in Table 3. The total benefit tb_S is calculated by: $tb_S = (Z_1 - Z_S) / Z_1$. Again, all data of Table 3 are averaged over 10 instances. For example: among our benchmark problems with $J = 2$ and 6 stages allowing 5 sublots, reduces the makespan to 54.76% compared to the situation if lot streaming is not applied.

Sublots Stages	*2*	*3*	*4*	*5*	*6*	*7*
3	25.44%	33.33%	36.77%	38.44%	39.37%	39.94%
4	29.38%	38.88%	43.42%	46.05%	47.71%	48.78%
5	33.19%	43.75%	48.57%	51.27%	53.30%	54.50%
6	34.69%	45.93%	51.48%	54.76%	56.89%	58.38%
7	35.57%	47.24%	53.11%	56.59%	58.82%	60.36%
8	36.52%	49.03%	55.18%	58.79%	61.18%	62.83%
9	38.81%	51.61%	57.92%	61.62%	64.05%	65.75%
10	39.04%	52.13%	58.60%	62.46%	65.01%	66.85%

Table 3: Total benefit of lot streaming with consistent intermingling sublots and $J = 2$, $M = \{3,..., 10\}$, $S = \{2,..., 7\}$

If the situation of multi product lot streaming with versus without intermingling sublots is considered, we found the following averaged percentage results (over 100 benchmark instances):

S	2	3	4	5	6	7
Mean	1.39	2.49	3.37	4.05	4.60	5.01
σ	2.51	3.57	4.71	5.71	6.39	6.97
Range: Min	0.00	0.00	0.00	0.00	0.00	0.00
Max	10.22	15.59	21.65	27.39	31.61	34.92

Table 4: Comparison of intermingling versus non-intermingling sublots and $J = 2$, $M = (3,..., 10)$, $S = (2,..., 7)$

On average, over 100 benchmark instances, lot streaming with intermingling is 5.01% better than lot streaming without intermingling, if seven sublots are allowed for each product. The standard deviation, σ, is 6.97% in this case. The minimal deviation is zero and the maximal deviation is 34.92%. This means that for at least one of the benchmark instances identical optimal schedules for lot streaming with and without intermingling exist. On the other hand there is a benchmark instance (2_6_4) where lot streaming with intermingling sublots gives an advantage of 34.92% over lot streaming without intermingling; the optimal makespan with and without intermingling is 435.74 and 587.93, respectively. Again the results for $J = 3$ are omitted here, as they show a similar pattern. The maximum deviation was found to increase with an increasing number of sublots, which is independent on the number of stages. As the mean deviation seems to be quite small, the application of non-intermingling sublots is a good recommendation for many instances, especially if setups have to be considered. Nevertheless, approaches to calculate solutions with intermingling sublots are valuable, as in some settings they may offer remarkable improvements; up to 34.92% for our benchmark instances.

4 Summary

Chang/Chiu (2005, p. 1532) recommend to tackle multiple product lot streaming problems not by hierarchical approaches but by simultaneous solution procedures. We have been able to present a model formulation to solve the multi-stage multi-product flow shop problem with sublots that are allowed to intermingle by standard optimization software. The applicability of the model formulation is due to the alleged complexity status of the problem and the subsequent use of binary variables somehow limited. However, we have been able to solve problems with 2 or 3 products and up to 7 sublots per product to optimality in a reasonable time. The number of stages hardly influences the effort to solving the problem; for instance solving a problem with 40 stages and 7 sublots per product takes less than 15 minutes.

From the computational results it became obvious that it is, at least for some instances, very beneficial to allow the sublots to intermingle in a multi-stage multi-product flow shop environment. Thus future research might be directed towards the development of meta heuristical solution approaches to solve larger instances of multiple product lot streaming problems; the application of meta heuristics for example is recommendable for integer lot sizes especially.

Acknowledgments

We wish to thank two anonymous referees for their helpful comments on an earlier version of this paper.

Literature:

Adams, J., Balas, E., Zawack, D.: The shifting bottleneck procedure for job shop scheduling. Management Science, Vol. 34 (1988), 391-401

Allahverdi, A., Gupta J.N.D., Aldowaisan, T.: A review of scheduling research involving setup considerations. Omega, Vol. 27 (1999), 219-239

Baker, K.R.: Lot streaming in the two-machine flow shop with setup times. Annals of Operations Research, Vol. 57 (1995), 1-11

Baker, K.R., Jia, D.: A comparative study of lot streaming procedures. Omega, Vol. 21 (1993), 561-566

Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.C.G., Stewart, W.R.: Designing and reporting on computational experiments with heuristic methods. Journal of Heuristics, Vol. 1 (1995), 9-32

Biskup, D., Feldmann, M.: Lot streaming with variable sublots: an integer programming formulation, Journal of the Operational Research Society, Vol. 57 (2006), 296-303

Chang, J.H., Chiu, H.N.: A comprehensive review of lot streaming. International Journal of Production Research, Vol. 43 (2005), 1515-1536

Dauzère-Pérès, S., Lasserre, J.B.: Lot streaming in job-shop scheduling. Operations Research, Vol. 45 (1997), 584-595

Feldmann, M.: Losüberlappung – Verfahren zur Effektivitätssteigerung in der operativen Produktionsplanung. Betriebswirtschaftliche Forschungsergebnisse, Band 132, Duncker & Humblot, Berlin, 2005 (in German)

Garey, M.R., Johnson, D.S., Sethi, R.: The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research, Vol. 1 (1976), 117-129

Hall, N.G., Sriskandarajah, C.: A survey of machine scheduling problems with blocking and no-wait in process. Operations Research, Vol. 44 (1996), 510-525

Hall, N.G., Laporte, G., Selvarajah, E., Sriskandarajah, C.: Scheduling and lot streaming in flowshops with no-wait in process. Journal of Scheduling, Vol. 6 (2003), 339-354

Hall, N.G., Laporte, G., Selvarajah, E., Sriskandarajah, C.: Scheduling and lot streaming in two-machine open shops with no-wait in process. Naval Research Logistics, Vol. 52 (2005), 261-275

Johnson, S.M.: Optimal two- and three-stage production schedules with setup time included. Naval Research Logistics Quarterly, Vol. 1 (1954), 61-68

Kalir, A.A.: Optimal and heuristic solutions for the single and multiple batch flow shop lot streaming problems with equal sublots. PhD Thesis, State University, Virginia (1999)

Kalir, A.A., Sarin, S.C.: Evaluation of the potential benefits of lot streaming in flow-shop systems. International Journal of Production Economics, Vol. 66 (2000), 131-142

Kalir, A.A., Sarin, S.C.: A near-optimal heuristic for the sequencing problem in multiple-batch flow shops with small equal sublots. Omega, Vol. 29 (2001), 577-584

Kalir, A.A., Sarin, S.C.: Constructing near optimal schedules for the flow-shop lot streaming problem with sublot-attached setups. Journal of Combinatorial Optimization, Vol. 7 (2003), 23-44

Kumar, S., Bagchi, T.P., Sriskandarajah, C: Lot streaming and scheduling heuristics for *m*-machine no-wait flowshops. Computers & Industrial Engineering, Vol. 38 (2000), 149-172

Lee, I., Sikora, R., Shaw, M.J.: Joint lot sizing with genetic algorithms for scheduling: evolving the chromosome structure. In: Forrest, S. et al. (eds.) Proceedings of the fifth International Conference on Genetic Algorithms. Morgan Kaufmann, 1993, 383-389

Lee, C.-Y., Lei, L., Pinedo, M.: Current trends in deterministic scheduling. Annals of Operations Research, Vol. 70 (1997), 1-41

Manne, A. S.: On the job-shop scheduling problem, Operations Research, Vol. 8 (1960), 219-223

Potts, C.N., Baker, K.R.: Flow shop scheduling with lot streaming. Operations Research Letters, Vol. 8 (1989), 297-303

Potts, C.N., Van Wassenhove, L.N.: Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity, Journal of the Operational Research Society, Vol. 43 (1992), 395-406

Şen, A., Benli, Ö.S.: Lot streaming in open shops. Operations Research Letters, Vol. 23 (1999), 135-142

Şen, A., Topaloğlu, E., Benli, Ö.S.: Optimal streaming of a single job in a two-stage flow shop. European Journal of Operational Research, Vol. 110 (1998), 42-62

Sriskandarajah, C. Wagneur, E.: Lot streaming and scheduling multiple products in two-machine no-wait flowshops. IIE Transactions, Vol. 31 (1999), 695-707

Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research, Vol. 64 (1993), 278-285

Trietsch, D., Baker, K.R.: Basic techniques for lot streaming. Operations Research, Vol. 41 (1993), 1065-1076

Vickson, R.G., Alfredsson, B.E.: Two- and three-machine flow shop scheduling problems with equal sized transfer batches. International Journal of Production Research, Vol. 30 (1992), 1551-1574

Vickson, R.G.: Optimal lot streaming for multiple products in a two-machine flow shop. European Journal of Operational Research, Vol. 85 (1995), 556-575

Zhang, W., Changyu, Y., Liu, J., Linn, R.J.: Multi-job lot streaming to minimize the mean completion time in m-1 hybrid flowshops. International Journal of Production Economics, Vol. 96 (2005), 189-200

Appendix

	number of sublots S						
instance	1	2	3	4	5	6	7
2_3_1	418	283	255.52	242.05	234.95	231.81	229.92
2_3_2	501	348.53	297.94	273.67	263.59	259.12	256.61
2_3_3	394	298.75	269.64	256.85	250.35	246.81	244.81
2_3_4	468	341.68	305.34	297.62	295.4	294.57	294.24
2_3_5	333	266.45	246.29	238.05	234.49	232.78	231.94
2_3_6	471	364.38	324.48	306.55	295.95	289.05	284.56
2_3_7	709	519.61	451.54	419.17	406.49	398.75	393.76
2_3_8	596	436	383.54	357.95	343.09	333.58	327.11
2_3_9	772	589.83	527.89	498.97	482.41	471.99	465.05
2_3_10	526	416.8	387.77	377.26	373.05	371.29	370.55
2_4_1	1142	828.5	724	670.45	635.55	615.08	604.57
2_4_2	792	541.53	460.01	420.68	398.16	384	374.55
2_4_3	1007	677.83	579.13	534.47	510.04	495.28	485.82
2_4_4	1146	811.31	697.07	640.91	608.43	585.96	570.52
2_4_5	540	402	351.88	331.36	319.64	312.7	308.74
2_4_6	638	446.17	391.66	361.11	344.01	332.67	325.69
2_4_7	1020	684.67	573.83	524.93	495.84	477.62	465.17
2_4_8	333	237	205	189	179.4	173	168.43
2_4_9	1221	897.25	798.07	754.33	732.15	719.96	712.99
2_4_10	974	691.3	603.7	560.06	535.37	519.07	508.06
2_5_1	1020	686.39	591.78	540	507	491.74	479.13
2_5_2	897	586.33	492.63	444.54	417.73	398.2	385.75
2_5_3	793	545.31	462.88	420.73	396.64	381.2	370.56
2_5_4	1476	948.48	756.95	687.25	658.5	639.67	626.39
2_5_5	889	611.33	537.12	512.74	503.76	500.09	498.47
2_5_6	1434	931.06	747.78	660.43	605.57	569.57	544.52
2_5_7	856	586.41	497.59	472.15	453.72	409.87	397.69
2_5_8	1077	705.03	612.65	566.62	539.27	521.11	508.21
2_5_9	1251	847	710.5	642.37	606.06	579.75	563.47
2_5_10	1026	687.95	572.11	512.95	482.14	466.34	456.54
2_6_1	1246	801.68	679.83	617.04	582.82	557.98	542.25
2_6_2	1967	1315.27	1099.47	992.2	929.23	886.52	857.51
2_6_3	1140	785.09	674.08	619.6	589.32	570.87	558.98
2_6_4	1008	682.68	568.23	510.16	475.22	452.69	435.74
2_6_5	1432	922.36	758.26	676.39	629.33	599.85	579.37
2_6_6	838	533.26	434.22	391.35	366.57	352.22	341.74
2_6_7	2257	1472.25	1218.37	1093.24	1019.43	970.43	935.69
2_6_8	1827	1114	876.25	756.02	684.34	636.15	600.47
2_6_9	1219	772.6	624.3	550.51	503.31	473.73	454.17
2_6_10	1406	951.22	804.02	732.54	689.54	662.92	643.32

Table 5: Optimal makespan of 2-job lot streaming instances with intermingling (Part I)

	number of sublots S						
instance	1	2	3	4	5	6	7
2_7_1	1458	927.44	790.38	729.27	697.28	680.01	668.9
2_7_2	1291	864.11	712.89	633.43	585.08	552.61	529.35
2_7_3	2508	1626.25	1334.89	1189.2	1100.89	1042.49	1000.96
2_7_4	1994	1244.96	994.43	867.18	791.17	741.24	705.97
2_7_5	1372	923.55	769.69	699.16	659.1	633.66	617.45
2_7_6	1436	923.24	753.01	661.02	609.28	576.68	558.01
2_7_7	619	402.78	331.09	293.18	271.86	257.7	247.27
2_7_8	1653	1049.38	849.96	751.57	693.58	655.77	629.46
2_7_9	1680	1103.74	912.45	815.93	757.1	718.51	691.23
2_7_10	1023	620.59	486.49	419.46	377.11	351.15	333.49
2_8_1	1525	917.5	715	619.12	561.76	523.93	498.87
2_8_2	1879	1168.95	913.51	783	707.7	659.66	624.98
2_8_3	2030	1266.97	1014.18	886.49	810.03	759.5	723.12
2_8_4	783	529.35	441.04	398.94	375.83	361.11	351.51
2_8_5	1158	762.78	623.62	557.7	517.79	491.99	475.19
2_8_6	2900	1792.75	1423.93	1236.24	1127.85	1058.7	1011.56
2_8_7	1400	872	700.57	617.8	569.19	537.16	514.56
2_8_8	2668	1727.48	1389.47	1224.37	1132.64	1069.15	1024.52
2_8_9	2421	1493.82	1170.27	1020.68	930.72	868.53	826.88
2_8_10	1096	722.86	601.3	538.43	499.55	473.15	453.93
2_9_1	1494	878.1	672.85	579	524.52	488.96	463.87
2_9_2	2427	1473.68	1157.67	998.1	907.42	845.73	801.93
2_9_3	1606	982.48	782.99	681.96	618.93	579.41	552.12
2_9_4	1974	1230.42	983.28	850.68	770.87	719.58	683.62
2_9_5	1892	1160.85	918.54	799.6	729.11	683.14	649.69
2_9_6	1363	870.56	705.06	625.26	576.84	545.8	523.33
2_9_7	2488	1467.29	1130.7	974.06	890.17	835.21	796.68
2_9_8	1681	1058.01	830.42	724.01	661.36	620.9	592.93
2_9_9	2507	1580.96	1270.18	1115.74	1023.31	961.15	918.27
2_9_10	2616	1535.25	1227.97	1067.02	974.48	907.92	866.99
2_10_1	1842	1079.37	826.14	700.94	627.52	578.26	543.75
2_10_2	2030	1253.7	994.22	852.9	771.21	718.17	679.92
2_10_3	1672	1022.32	807.64	698.57	632.74	589.28	558.12
2_10_4	2682	1641.01	1260.95	1075.11	968.64	897.58	844.17
2_10_5	1460	906.09	721.8	639.58	592.93	563.23	541.74
2_10_6	2216	1412.04	1119.6	980.59	892.99	835.95	795.11
2_10_7	2396	1460.21	1143.42	985.12	885.88	820.42	772.61
2_10_8	2348	1415.16	1085.75	925.64	827.91	763.47	717.63
2_10_9	1248	740.27	592.1	519.59	476.12	447.16	426.54
2_10_10	1754	1063.24	847.1	737.75	673.17	630.79	600.01

Table 6: Optimal makespan of 2-job lot streaming instances with intermingling (Part II)

	number of sublots S			
instance	1	2	3	4
3_3_1	636	540	512.57	501.6
3_3_2	537	454.06	433.29	425.9
3_3_3	1102	782.58	744.65	742.48
3_3_4	769	620.2	573.21	551.52
3_3_5	976	838.22	806.29	797.81
3_3_6	949	838.85	819.71	816.78
3_3_7	713	596	572.73	566.62
3_3_8	1039	875.67	840.78	832.27
3_3_9	640	571.31	561.53	560.25
3_3_10	1202	981.15	920.75	894.74
3_4_1	1213	896.38	808.94	772.79
3_4_2	1210	951.02	879.98	849.04
3_4_3	686	495	470.76	465.31
3_4_4	920	728.61	666.85	639.31
3_4_5	1086	831.49	739.76	693.89
3_4_6	833	622.97	555.54	526.18
3_4_7	1885	1412.63	1271.98	1209.12
3_4_8	1634	1114.38	941.23	854.7
3_4_9	792	611.57	554.81	531.88
3_4_10	1512	1082.37	955.59	901.55
3_5_1	2148	1489.69	1271.7	1158.93
3_5_2	1515	1035.11	886.65	818.14
3_5_3	1744	1190.64	1001.56	910.72
3_5_4	1325	892.27	748.21	675.62
3_5_5	1770	1251.66	1128.19	1073.24
3_5_6	1953	1342.54	1174.18	1114.82
3_5_7	1244	884.53	778.17	727.23
3_5_8	2256	1675.08	1491.59	1412.51
3_5_9	1277	921.67	808.39	753.78
3_5_10	640	449.6	387.63	357.73
3_6_1	1459	974.03	825.72	753.72
3_6_2	1756	1184.41	993.02	915.23
3_6_3	966	716.14	636.67	600.51
3_6_4	1590	1089.87	927.98	843.58
3_6_5	1516	1070.02	925.67	853.74
3_6_6	1346	901.4	763.4	694.81
3_6_7	2003	1308.52	1095.55	989.2
3_6_8	1169	803.38	685.52	625.91
3_6_9	1885	1285.34	1081.89	989.45
3_6_10	1415	983.52	833.81	764.21

Table 7: Optimal makespan of 3-job lot streaming instances with intermingling (Part I)

	number of sublots S			
instance	1	2	3	4
3_7_1	1730	1196.96	999.83	899.28
3_7_2	1989	1233.87	993.9	888.42
3_7_3	919	635.77	546.17	505.62
3_7_4	2145	1415.86	1187.77	1077.98
3_7_5	1524	1046.44	888.26	808.77
3_7_6	1029	693.08	582.07	526.29
3_7_7	2222	1419.97	1152.69	1019.09
3_7_8	1792	1364.78	1172.42	1079.66
3_7_9	1130	764.02	642.96	588.16
3_7_10	1518	970.44	810.68	751.17
3_8_1	1962	1190.82	945.43	822.84
3_8_2	2003	1274.17	1027.31	905.64
3_8_3	1670	1031	819.18	713.06
3_8_4	2610	1692.11	1386.23	1240.26
3_8_5	1154	700.6	562.08	497.47
3_8_6	2057	1297.82	1044.9	918.54
3_8_7	2360	1576.63	1311.92	1178.54
3_8_8	2170	1310.77	1106.95	1029.69
3_8_9	2110	1449.55	1243.06	1143.97
3_8_10	1571	1048.5	866.32	784
3_9_1	1363	887.72	725.36	699.06
3_9_2	1184	759.17	614.74	542.68
3_9_3	2239	1465.72	1202.65	1074.49
3_9_4	1602	1013.8	824.98	730.04
3_9_5	2772	1692	1335.23	1165.02
3_9_6	1276	829.98	687.86	620.6
3_9_7	2209	1548.39	1322.92	1201.34
3_9_8	2053	1242.35	991.18	871.11
3_9_9	2161	1317.55	1045.38	908.45
3_9_10	2320	1420.93	1112.43	962.84
3_10_1	3087	1916.54	1570.87	1408.19
3_10_2	1864	1125.92	909.03	796.77
3_10_3	2325	1412.87	1100.29	959.9
3_10_4	2865	1806.4	1457.62	1281.24
3_10_5	2540	1621.28	1306.11	1145.8
3_10_6	3288	1919.44	1460.06	1232.45
3_10_7	2218	1413.33	1141.84	1013.7
3_10_8	2500	1613.81	1314.8	1179.46
3_10_9	1419	830.54	651.45	566.29
3_10_10	2303	1433.23	1142.21	995.99

Table 8: Optimal makespan of 3-job lot streaming instances with intermingling (Part II)

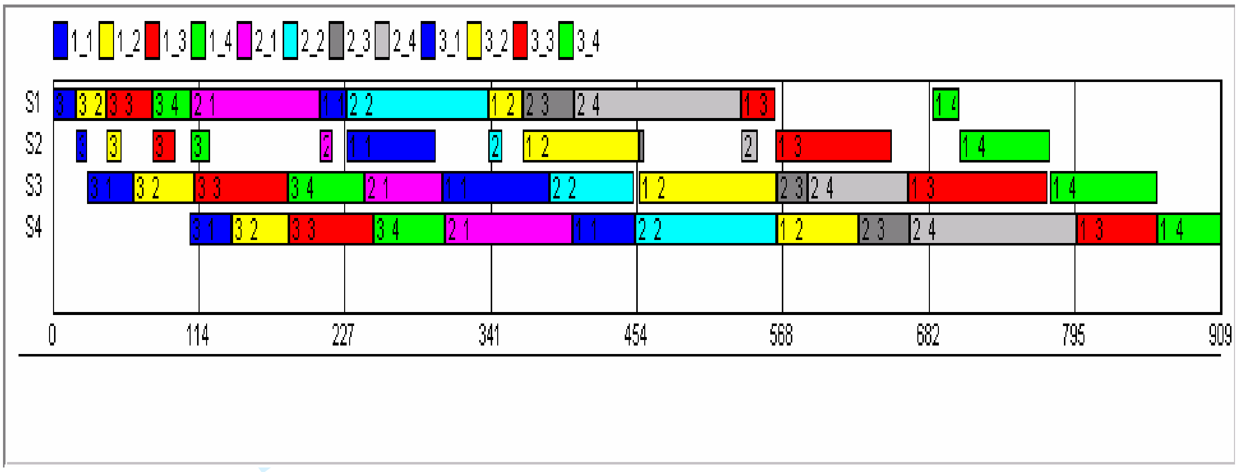

Figure 1: Optimal solution of instance 3_4_10 with S = 4 intermingling discrete sublots

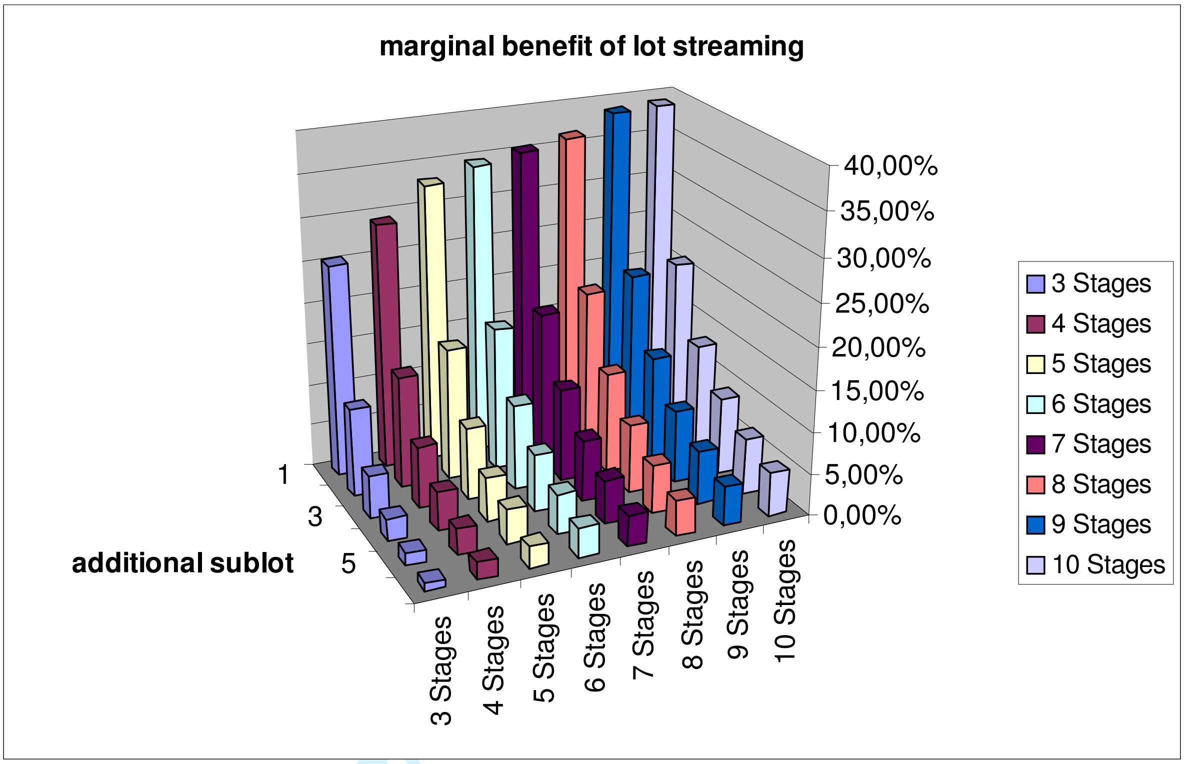

Figure 2: Marginal benefit of lot streaming with consistent intermingling sublots and $J = 2$, $M = \{3,..., 10\}$, $S = \{1,.., 7\}$