

Heterogeneous batch structures in throughput scheduling

P.J. Weeda

Department of Mechanical Engineering, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

Recently a few papers appeared on throughput scheduling, dealing with the relationship between batch structure and process structure in discrete batch production, while maximizing time-constrained throughput. Results have been concentrated on the class of homogeneous batch structures, i.e. batch structures with equal batch sizes for each process per cycle.

In this paper heterogeneous batch structures are considered. By numerical examples, it is shown that heterogeneous batch structures can outperform the best result obtained by considering only homogeneous batch structures. Moreover integer programs are developed, which generate such solutions.

1. Introduction

At present there is a considerable interest in batching and lot streaming in relation to scheduling. Different from classical lot sizing procedures (balancing inventory holding costs against set-up costs) the aim is to maximize throughput, while keeping a reasonable due date performance. This is a principle dilemma met in many industries: large batches yield efficient capacity utilization, but in order to meet due dates, it makes sense to switch to a new operation in time.

Some references on this subject are Refs. [1–3]. Karmakar, Kekre and Kekre [1] have dealt with the relationship between lot sizing and job flow times by examining the impact of lot sizes on flow times using a simulation model and Q-lots. Baker and Pyke [2] present a computationally efficient procedure for the m -machine, two-sublot problem and discuss computational results for heuristic approaches

to the more than two sublot problem. A general framework combining batching and lot sizing with scheduling is developed in a recent paper by Potts and van Wassenhove [3].

This paper is concerned with the interesting question, which kind of batch structure maximizes time-constrained throughput in a particular process structure. The process structure considered is serial and consists of $n + 1$ processes, where n is an arbitrary, but finite natural number. It is described as follows.

A material is successively subjected to n processes on a facility A and one finishing process on facility B. In our terminology facility A is called a multi-process facility and facility B a one-process facility. For each process a set-up time and a unit processing time are specified. The objective is to maximize time-constrained throughput, i.e. the number of units of finished product produced within a given time allowance T .

Two additional assumptions are made. The transfer batch between the two facilities A and B equals one unit of product and material stock is sufficient to obtain a maximum

Correspondence to: P.J. Weeda, Department of Mechanical Engineering, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

through-put during T . Throughout the paper the word time-constrained will be omitted for convenience.

In previous papers [4–7], attention has been paid to the question which batch structure maximizes throughput. Distinction has been made between single batch and multiple batch structures. In the case of a single batch structure there is one common batch size for each process. The throughput is equal to this common batch size. If for at least one process at least two batches are processed then a multiple batch structure is used.

A batch cycle consists of one batch for each process. If the sizes of these batches are all equal, then the batch cycle is called homogeneous. A homogeneous multiple batch structure consists of two or more homogeneous batch cycles. In this case the batch sizes are equal per batch cycle, but may be different for different batch cycles. A heterogeneous multiple batch structure permits splitting of the batch size of a homogeneous batch cycle in order to achieve an additional reduction of idle time on the second facility. Note that a multiple batch structure as well as a further splitting of batch sizes are beneficial due to the assumption of the full availability of both facilities during the time allowance T .

Maximum throughput for the class of homogeneous batch structures can be obtained by integer programming. For each number of homogeneous batch cycles $m = 1, 2, \dots$ an integer program can be formulated for some process structure. Such an integer program has m constraints. Each constraint reflects a critical utilization path, i.e. a path of full utilization during the time allowance T , cf. Refs. [5, 6]. Primarily one is interested in whether a single or a multiple batch structure is optimal. In Ref. [7] an overview of results in this respect is given. For certain process structures a single batch structure appears to be optimal, meaning that, by induction on the number of homogeneous batch cycles given by m , the maximum throughput is proven to be obtained for $m = 1$. In other structures (among those the one explored in this paper) a single batch structure is not optimal, because a double batch structure

is proven to yield better results. However the question remains, whether homogeneous batch structures are truly optimal. This paper provides examples for which heterogeneous batch structures outperform the best homogeneous ones. Such examples may be obtained by the integer programs developed in this paper. The dual formulations of these integer programs consider minimization of the make-span for a given total production quantity Q . As such they fit into the general framework presented in Ref. [3].

2. Homogeneous batch structures

Primarily some additional notation is introduced. The process structure in the introduction can be represented by $(A_1, A_2, \dots, A_n, B)$, where A_j denotes process j on facility A for $j = 1(1)n$. The finishing process on facility B is simply denoted by B. Set-up times and unit processing times are represented by $S(\cdot)$ and $P(\cdot)$ respectively. The set-up time for process B may be assumed to be zero without much loss of generality. The batch size of batch cycle i is denoted by Q_i , $i = 1(1)m$. The maximum throughput for m batch cycles is denoted by $MT(m)$.

Starting with $n = 2$, i.e. process structure (A_1, A_2, B) , some previously obtained results are summarized (cf. Refs. [5, 6]). Distinction is necessary between two cases: $P(B) \leq P(A_2)$ and $P(B) > P(A_2)$. For $P(B) \leq P(A_2)$ a single batch structure is optimal. For $P(B) > P(A_2)$ a multiple batch structure is optimal since it can be proven that $MT(2) \geq MT(1)$ (cf. Ref. [6]). The integer programs for $MT(1)$ and $MT(2)$ in the case $P(B) > P(A_2)$ are respectively given by

$$\max MT(1) = Q,$$

subject to

$$S(A) + QP(A_1) + P(A_2) + QP(B) \leq T,$$

$$Q \geq 0 \text{ and integer,}$$

and

$$\max MT(2) = Q_1 + Q_2,$$

subject to

$$2S(A) + Q_1P(A) + Q_2P(A1) + P(A2)$$

$$+ Q_2P(B) \leq T,$$

$$S(A) + Q_1P(A1) + P(A2) + (Q_1 + Q_2)P(B) \leq T,$$

$$Q_1, Q_2 \geq 0 \text{ and integer,}$$

where $P(A) = P(A1) + P(A2)$ and $S(A) = S(A1) + S(A2)$. Each constraint of these integer programs reflects a critical utilization path. A critical utilization path is defined as a path of full utilization during the time allowance T , cf. Refs. [5, 6]. For $MT(m)$, $m = 3, 4, \dots$, integer programs can be formulated, which yield the maximum throughput for m homogeneous batch cycles. The function $MT(m)$ appears to attain a maximum for some m .

The integer programs for process structure (A1, A2, B) can be easily extended to n processes on facility A by replacing $P(A1)$ by $P(A1) + P(A2) + \dots + P(A_{n-1})$ and $P(A2)$ by $P(A_n)$ and defining $P(A) = P(A1) + \dots + P(A_n)$ and $S(A) = S(A1) + \dots + S(A_n)$. In fact the integer programs for an n -process facility are equivalent to those for a 2-process facility. Hence the proof that $MT(2) \geq MT(1)$, given in Ref. [6] for process structure (A1, A2, B) is also valid for process structure (A1, A2, ..., A $_n$, B).

Next a numerical example is given for $n = 4$. The data are:

$$P(A1) = P(A2) = P(A3) = P(A4) = 10,$$

$$P(B) = 50,$$

$$S(A1) = S(A2) = 50, \quad S(A3) = S(A4) = 20,$$

$$T = 2000.$$

The results for $m = 1(1)6$ are given by

$$MT(1) = 23, \quad MT(2) = 28, \quad MT(3) = 30,$$

$$MT(4) = 30, \quad MT(5) = 30, \quad MT(6) = 28.$$

In the next section it will be shown that a solution with a throughput of 31 exists. This solution has a heterogeneous multiple batch structure with six batch cycles.

3. Heterogeneous batch structures

A heterogeneous multiple batch structure permits variation in batch size per cycle. In this section heterogeneous batch structures are considered, which are constructed by applying a batch splitting procedure to a homogeneous batch cycle. The common batch size for each process Q of a homogeneous batch cycle will be called a throughput batch size, since it may be considered as a split of the total throughput. In turn the throughput batch size is split into at least two subcycles. For each subcycle batch sizes are specified for each process. If a process is not used in a subcycle, the corresponding batch size is equal to zero. The batch sizes can be arranged in a matrix with processes and subcycles as entries. All columns in this matrix sum up to the throughput batch size Q .

In the case of a serial process structure for which the ingoing material is successively processed in batches by each process of the process structure, a restriction should be put on the subcycles. For process structure (A1, A2, ..., A $_n$, B) this restriction states that for each process $j = 2(1)n + 1$ in the sequence, the total number of units produced by the foregoing process should be sufficient to produce its prescribed batch size. It implies for example that the first row of the process-subcycle matrix is non-increasing. In general the batch size q_{ij} of process j in subcycle i should satisfy

$$q_{ij} \leq \sum_{k=1}^i q_{k,j-1},$$

$$\text{for } i = 1(1)m, \quad j = 2(1)n + 1, \quad (1)$$

and

$$\sum_{i=1}^m q_{ij} = Q, \quad \text{for } j = 1(1)n + 1, \quad (2)$$

where m is the number of subcycles. In Fig. 1 only the batch structures (a) and (b) exhibit feasible subcycles.

One important reason for considering heterogeneous batch structures might be the distribution of the set-up times of the processes. The numerical example presented in the preceding section with $S(A1) = S(A2) > S(A3) = S(A4)$ suggests to treat processes A1 and A2 differently from processes A3 and A4. If process-subcycle matrix would be

$$\begin{pmatrix} q+r & q+r & q & q \\ 0 & 0 & r & r \end{pmatrix} \quad (3)$$

for a throughput batch size of $Q = q + r$. Note that only the processes A3 and A4 are involved in the split of the throughput batch.

On the other hand it can be demonstrated that it makes no sense to consider heterogeneous batch cycles if the number of processes on facility A is equal to two ($n = 2$). The integer program for one heterogeneous batch cycle with process-subcycle matrix given by

$$\begin{pmatrix} q+r & q \\ 0 & r \end{pmatrix} \quad (4)$$

yields two constraints given by

$$S(A) + (q+r)P(A1) + P(A2) + (q+r)P(B) \leq T,$$

$$S(A) + (q+r)P(A1) + P(A2) + S(A2)$$

$$+ qP(A2) + rP(B) \leq T.$$

a)	9 9 4 4 4	b)	4 4 4 0 0
	0 0 5 5 5		5 5 5 9 9
c)	0 0 5 5 5	d)	4 4 4 9 9
	9 9 4 4 4		5 5 5 0 0

Fig. 1.

The first constraint refers to the critical utilization path, where the switch from facility A to facility B occurs after the first unit of subbatch q and remains on B thereafter. The second constraint refers to the second possible critical utilization path, where the switch occurs after the first unit of subbatch r . In both cases there is an initial idle time on facility B which is equal to $S(A) + (q+r)P(A1) + P(A2)$. In the second case there is an additional idle time on facility B between the two subbatches q and r , which is equal on facility B between the two subbatches q and r , which is equal to $S(A2) + qP(A2) - qP(B)$. If this quantity is non-positive then both critical utilization paths coincide, since the latter idle time vanishes. If it is positive then the second critical utilization path exceeds the first one in time for fixed q and r , implying a smaller time-constrained throughput. The same reasoning can be applied to multiple throughput batches.

To investigate the numerical example of the preceding section further, integer programs are formulated with which splits of the kind described above, can be evaluated numerically. Primarily an integer program is developed to obtain the maximum throughput for a single throughput batch with the process-subcycle matrix given by (3). In this case there appears to be two possible critical utilization paths. The difference between them depends on the existence of idle time between the two batches of size q and r on facility B. If there is no idle time between them, the critical utilization path is on facility A from time zero until the first unit of batch q of process A4 is produced. Then it switches to facility B and remains on that facility until the end of the time allowance T . If there is an idle time between the two batches q and r then the second possible critical utilization path switches from facility A to facility B after production of the first unit of batch r of process A4. The corresponding Gantt chart is presented in Fig. 2.

The corresponding integer program is given by

$$\max MT(1) = Q = q + r,$$

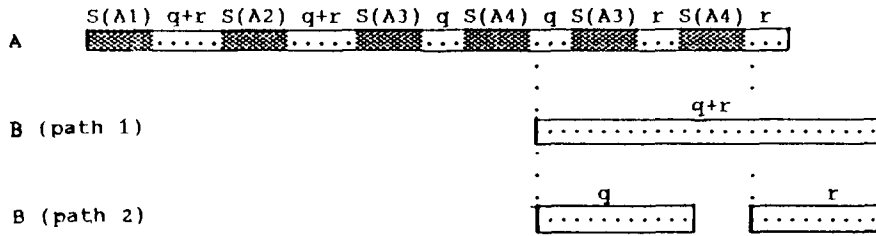


Fig. 2.

subject to

$$S(A) + (q + r)P(A12) + qP(A3) + P(A4) + (q + r)P(B) \leq T,$$

$$S(A) + (q + r)P(A12) + qP(A34) + S(A34) + rP(A3) + P(A4) + rP(B) \leq T,$$

$q, r \geq 0$ and integer.

In this program the following additional notation is used:

$$S(Ajk) = S(Aj) + S(Aj + 1) + \dots + S(Ak)$$

and

$$P(Ajk) = P(Aj) + P(Aj + 1) + \dots + P(Ak).$$

The integer program for two throughput batches $Q_1 = q_1 + r_1$ and $Q_2 = q_2 + r_2$, each with a process-subcycle matrix of the type given by (3), is formulated by

$$\max MT(2) = Q_1 + Q_2 = q_1 + r_1 + q_2 + r_2,$$

subject to

$$S(A) + (q_1 + r_1)P(A12) + q_1P(A3) + P(A4) + (q_1 + r_1 + q_2 + r_2)P(B) \leq T,$$

$$S(A) + (q_1 + r_1)P(A12) + q_1P(A34) + S(A34) + r_1P(A3) + P(A4) + (r_1 + q_2 + r_2)P(B) \leq T,$$

$$2S(A) + (q_1 + r_1)P(A) + S(A34) + (q_2 + r_2)P(A12)$$

$$+ q_2P(A3) + P(A4) + (q_2 + r_2)P(B) \leq T,$$

$$2S(A) + (q_1 + r_1)P(A) + 2S(A34) + (q_2 + r_2)P(A12)$$

$$+ q_2P(A34) + r_2P(A3) + P(A4)$$

$$+ r_2P(B) \leq T,$$

$q_1, r_1, q_2, r_2 \geq 0$ and integer.

Obviously this program reflects four possible critical utilization paths, each corresponding to one of the four possible switches from facility A to facility B after the production of the first unit of a batch of process A4.

The integer program for three throughput batches with sizes $Q_1 = q_1 + r_1$, $Q_2 = q_2 + r_2$ and $Q_3 = q_3 + r_3$ respectively, is developed along the same lines and will be omitted here.

The numerical results obtained by applying these three integer programs to our example are

$$MT(1) = 25, \quad MT(2) = 29, \quad MT(3) = 31.$$

By explicit enumeration there appear to be three solutions with $MT(3) = 31$. They are given by

13 13 4 4	13 13 4 4	13 13 4 4
9 9	9 9	9 9
11 11 3 3	11 11 3 3	11 11 3 3
8 8	8 8	8 8
7 7 4 4	7 7 3 3	7 7 2 2
3 3	4 4	5 5

Alternatively three instead of two processes can be involved in the splitting. The process-subcycle matrix then becomes

$$\begin{pmatrix} q+r & q & q & q \\ 0 & r & r & r \end{pmatrix}. \tag{5}$$

The integer program for one throughput batch becomes

$$\max MT(1) = Q = q + r,$$

subject to

$$S(A) + (q + r)P(A1) + qP(A23) + P(A4)$$

$$+ (q + r)P(B) \leq T,$$

$$S(A) + (q + r)P(A13) + qP(A4) + S(A24)$$

$$+ P(A4) + rP(B) \leq T,$$

$$q, r \geq 0 \text{ and integer.}$$

A comparison of this integer program with the one presented above for two processes involved in the split reveals that in the first constraint a term $rP(A2)$ is omitted and in the second constraint a term $S(A2)$ is added. It can also be shown that for two throughput batches

$Q_1 = q_1 + r_1$ and $Q_2 = q_2 + r_2$ in the left members of the four constraints respectively $-r_1P(A2)$, $S(A2)$, $-r_2P(A2)$ and $S(A2)$ are added.

A third possibility is to vary the number of processes involved in the splitting over the throughput batches. For example, if three throughput batches are considered, the processes A3 and A4 may be involved in the splitting of the first two throughput batches and the processes A2, A3 and A4 in the third throughput batch. Numerical results on these splitting devices applied to our example, are summarized in the next section.

4. Numerical results

Splits are denoted by a 3-tuple (a, b, c) , a , b and c specifying respectively the number of processes involved in the split of the first, the second and the third throughput batch. A zero stands for an unsplit throughput batch. If one process is involved in the split of a throughput batch then it will be process A4. The involvement of two processes refers to the processes A3 and A4. The involvement of three processes refers to the processes A2, A3 and A4. For example, the 3-tuple $(2, 2, 3)$ indicates that the processes A3 and A4 are involved in the splits of the first and the second throughput batch, while processes A2, A3 and A4 are involved in the split of the third throughput batch.

In Table 1, results are given for the maximum throughput obtained for respectively for

Table 1

	MT(1)	MT(2)	MT(3)
(0, 0, 0)	23	28	30
(1, 1, 1)	23	28	29
(1, 1, 2)	23	28	30
(1, 2, 2)	23	29	30
(2, 2, 2)	25	29	31 (3)
(2, 2, 3)	25	29	31 (6)
(2, 3, 3)	25	30	31 (5)
(3, 3, 3)	27	30	31 (2)

one, two and three throughput batches. In the case of one throughput batch the split is indicated by a . For two throughput batches the splits are indicated by a and b respectively. For three throughput batches the splits are indicated by a , b and c respectively.

The numerical results for one, two and three unsplit throughput batches have already been given in Section 2. For comparison they are included in Table 1 and denoted by the 3-tuple (0, 0, 0).

The number of solutions for the splits with maximum throughput of 31 found for this example, is given by the number between parentheses. The total number of different solutions is 16.

5. Conclusions

Since only one numerical example is investigated no general conclusions can be drawn. On the basis of this numerical example one may conclude:

(1) Splitting of throughput batches is worthwhile, although there is only a small increase of 1 with respect to the best result on homogeneous batch cycles of 30. Moreover, one should recognize that splitting leads to an increase of administration.

(2) The solutions with throughput 31 are obtained by splits involving 2 or 3 processes on facility A. One may conclude that splits involving only one process are not worthwhile.

(3) Regarding the results involving 2 or 3 processes in the split, (2, 3, 3) and (3, 3, 3) have the best overall performance. So the distribution of set-up times does not provide a clear guidance in deciding which processes should be included in the split.

Also for more complicated process structures integer programs for throughput batch splits can be developed. For example one may increase the number of one-process facilities or consider structures with alternating a multi-process facility and a number of one process facilities. It has been proven by means of equivalence of integer programs for these process structures that a single throughput batch (or equivalently a single batch structure) is not optimal, cf. Ref. [7]. For these process structures integer programs can be formulated and used in a similar way as is done in this paper.

References

- [1] Karmarkar, U.S., Kekre, S. and Kekre, S., 1985. hLotsizing in multi-item, multi machine job shops, *IIE Trans.*, 17(3): 290–298.
- [2] Baker, K.R. and Pyke, D.F., 1990. Solution procedures for the lot streaming problem, *Decision Sci.*, 21: 475–491.
- [3] Potts, C.N. and Van Wassenhove, L.N., 1992. Integrating scheduling with batching and lotsizing: a review of algorithms and complexity, *J. Op. Res. Soc.*, 43(5): 395–406.
- [4] Weeda, P.J., 1991. On the relation between batch mode, bottleneck machines and maximum throughput. In: M. Pridham and C. O'Brien (Eds.) *Production Research Approaching the 21st Century* Taylor & Francis, London, pp. 283–289.
- [5] Weeda, P.J., 1991. Maximizing throughput by evaluating critical utilization paths, *Int. J. Prod. Econ.*, 24: 191–198.
- [6] Weeda, P.J., 1992. Multiple batch structures in throughput scheduling, *Int. J. Prod. Econ.*, 26: 359–366.
- [7] Weeda, P.J., 1991. Throughput scheduling: an overview of present results, *Proc. 22nd Annual Meeting of the Midwest Decision Sciences Institute*, May 1–3, 1991. Indianapolis, Indiana.