Multi-product Batch Processing Time Maximization Problem

Abstract

In the plastic bag extrusion process, it is necessary to determine the optimal processing time for batches formed by different products, which are processed simultaneously by the same extruder, but with different processing rates. The batch processing time must be determined in order to meet a series of known constraints, such as the limitation for the quantity produced for each product and for the quantity produced for the set of all products in the same batch. In this paper we present this problem as a new combinatorial optimization problem named Multi-product Batch Processing Time Maximization (MPBPTM) problem. We also present a mathematical model for the MPBPTM problem and an analytical solution method with polynomial time complexity, which proved to be able to obtain optimal solutions for several benchmarks in a very short time, even for very large instances.

Keywords: multiproduct batch, processing time maximization, analytical solution, C++, LINGO

1. Introduction

The scientific literature presents several papers that address multi-product batch problems. Many works target the problem of multi-product batch scheduling (MPBS), in which it is necessary to determine the production sequence of the products and the ideal batch size for each product, in a production of several products which share the same production facilities and for which demand is known (Eilon, 1985). In this problem, a single product is processed at a time, so it is necessary that the quantity produced is sufficient to meet the demand during the period in which other products are being processed. The solution must be determined in such a way that both setup and storage costs are minimized. Among the authors who addressed this problem are Eilon (1985), Omega Journal (1993) and Liu *et al.* (2020).

Méndez *et al.* (2000) and Shi *et al.* (2017) present two parallel multi-product batch scheduling (PMPBS) problems, in which production units of the same type are arranged in parallel, forming each production stage. In this case, it is also necessary to define in which units each product will be processed. The authors also take into account other constraints for the problem (*e.g.*, release time and due date for the products, and sequencing constraints due to mismatched product colors sequences) and other goals (*i.e.*, optimize customer satisfaction and/or the plant performance). In the papers mentioned above, production in a single unit or in a serial flowshop plant is considered, in which all products follow a linear flow through the production stages. Kim *et al.* (1996) consider the multi-product batch problem in networked processes. As defined by the authors, in this case, the production units are not arranged in lines, and the connection between the units may or may not exist. Therefore, the production paths in the flowshop network can be different for each product. Also the passing of previous batch can be possible so that the final sequence of the production at the last stage units can be different from the initial sequence. Other important works deal with the problem of a multi-product batch plant design (MPBPD). In this case the problem is to obtain the configuration of the plant and the equipment that minimize the capital cost of all the equipment items needed to fulfill the production requirements (Ravemark and Rippin, 1998).

In this work, we are concerned with determining the ideal processing time for a batch consisting of a set of products, each product being or not different from the other products of the same batch. That is, we consider a specific case where different products are processed simultaneously on the same processor. This case occurs, for example, during the processing of plastic bags in extruders, where the resins are melted, then pass through a cylinder forming a balloon and this balloon, after cooling, is stored in a coil which, in turn, is cut into several smaller coils with different models of plastic bags.

As scientific contributions, in this paper we define a new multi-product batch problem, named multi-product batch processing time maximization (MPBPTM) problem, as well as a mathematical model and a polynomial time complexity analytical solution method for solving this problem.

In the next sections, a presentation of problem is made along with an application example. Section 3 presents an interger linear mathematical model for the problem and section 4 presents an analytical method for its solution. Section 5 presents the tests and results obtained by a solver in C++ applying the analytical solution and a solver developed in LINGO to new proposed benchmarks. In sections 6 and 8, the contributions to this work are informed and acknowledgments are given, respectively. Finally, section 7 presents the paper's conclusions and suggestions for future works.

2. Multi-product batch processing time maximization problem

The multi-product batch processing time maximization problem arises when a set of different products are processed simultaneously in a same production batch. In this problem, it is considered that the quantity produced of each product is directly proportional to the processing time, however, with a different constant of proportionality (production rate) for each product. In addition, there is a maximum quantity allowed for the production of batch products, defined both individually and for the set. The maximum production quantity of each product is mainly defined according to the demand for the product. However, it is still possible to stock the products and/or send them to the outlets. In both cases, there is a stocking/shipping limit for each product and a stocking/shipping limit for the set of products in the batch. Also, there is a time limit available for processing the batch. The problem consists of defining the maximum processing time for the batch, respecting the limitations related to the quantities produced. For a better understanding of the problem, an example is presented below.

Example: A certain machine must process a batch containing 2 different products: A and B. The production rate of A is 60 g/min while the production rate of B is 40 g/min. The factory has free stock for a maximum of 3000 g of any product, and, according to the company's inventory policy, a maximum of 3000 g of product A and 2000 g of product B can be stocked at the factory. There is a demand for 1000 g of product A and 500 g of product B. The factory has an outlet that has free space in stock of 1000 g, which can receive a maximum of 600 g of each product. A maximum time of 100 minutes of this machine can be allocated for processing this batch. What is the maximum possible time for processing this batch ?

3. Mathematical model

Given that:

UD_i is the demand for the product i;

I is the maximum quantity allowed for additional factory storage of all products in the batch;

UI_i is the maximum quantity allowed for stocking the product i in the factory;

O is the maximum quantity allowed for shipment of all products to outlets;

UO_i is the maximum amount of product i that can be shipped to outlets;

p_i is the production rate of product i;

Z is the timeout for batch processing;

P_i is the amount of product i produced;

D_i is the amount of product i delivered for the demand;

O_i amount of product i shipped to factory outlets;

I_i is the amount of product i that will be stored at the factory;

T is the batch processing time.

We have the problem:

$$max \quad T \tag{1}$$

s.t.

$$P_i - \mathrm{p}_i * T = 0 \quad \forall i \tag{2}$$

$$P_i - D_i - O_i - I_i = 0 \quad \forall i \tag{3}$$

$$D_i \leq \mathrm{UD}_i \quad \forall i \tag{4}$$

$$O_i \leq \mathrm{UO}_i \quad \forall i \tag{5}$$

$$\sum_i O_i \leq \mathrm{O} \tag{6}$$

$$I_i \leq \mathrm{UI}_i \quad \forall i \tag{7}$$

$$\sum_i I_i \leq \mathrm{I} \tag{8}$$

$$T \leq \mathrm{Z} \tag{9}$$

$$T, D_i, O_i, I_i \in \mathbb{Z}^+ \quad \forall i \tag{10}$$

where:

Constraints in (2) relate the quantity produced, P_i, to batch processing time T. Constraints in (3) calculate the quantity produced, P_i, as a function of the primary variables, D_i, O_i and I_i. Constraints in (4), (5), and (7) state that the quantity delivered to demand, the quantity shipped to the autlets, and the factory-stocked quantity of each product must be less than their respective known limits. Constraints (6) and (8) state that both the sum of product quantities sent to the autlets and the sum of product quantities stocked in the factory must be less than their respective maximum allowed values. The restriction in (9) establishes that there is a batch processing time limit, Z, that must be respected. And finally, the constraints in (10) inform the nature of the decision variables.

4. Analytical solution

It is possible to consider the factory stock and the outlets stock as single stock, so we have:

$$E_i = O_i + I_i \tag{11}$$

where E_i is the sum of the quantity stored at the factory and the quantity sent to the outlets of the product i.

So that:

$$E_i \leq \mathrm{UO}_i + \mathrm{UI}_i \tag{12}$$

and

$$\sum_i E_i \leq \mathrm{O} + \mathrm{I} \tag{13}$$

It is also possible to split the batch processing time into two time slots:

$$T = T' + T'' \tag{14}$$

and consider that T' is the maximum processing time used only for production that will meet the demand. Thus, we can find T', solving the reduced problem:

$$max \quad T' \tag{15}$$

$s.t.$

$$D_i - \mathrm{p}_i * T' = 0 \quad \forall i \tag{16}$$

$$D_i \leq \mathrm{UD}_i \quad \forall i \tag{17}$$

$$D_i, T' \in \mathbb{Z}^+ \quad \forall i \tag{18}$$

This reduced problem can be rewritten in the form:

$$max \quad T' \tag{19}$$

$s.t.$

$$T' \leq \lfloor \mathrm{UD}_i/\mathrm{p}_i \rfloor \quad \forall i \tag{20}$$

So we have that T' will be the smallest of the ratios $\lfloor \mathrm{UD}_i/\mathrm{p}_i \rfloor$ of all products.

Once we find the value of T', we can calculate the value of unmet demand for each product after T', S_i, through the equations in (21).

$$S_i = \mathrm{UD}_i - \mathrm{p}_i * T' \quad \forall i \tag{21}$$

Now we consider that the time interval T'' will be used for the production of the quantities that will be stored (in the factory and in the outlets), as well as of the demand not met by the production in the first time interval, S_i.

In this case, we can find T''' by solving the second reduced problem:

$$max \quad T'' \tag{22}$$

s.t.

$$E_i - \mathrm{p}_i * T'' = 0 \quad \forall i \tag{23}$$

$$E_i - S_i \leq \mathrm{UO}_i + \mathrm{UI}_i \quad \forall i \tag{24}$$

$$\sum_i E_i - S_i \leq \mathrm{O} + \mathrm{I} \tag{25}$$

$$E_i, T'' \in \mathbb{Z}^+ \quad \forall i \tag{26}$$

Again, using a little algebra, we can rewrite this reduced problem in the form:

$$max \quad T'' \tag{27}$$

s.t.

$$T'' \leq \lfloor (\mathrm{UO}_i + \mathrm{UI}_i + S_i)/\mathrm{p}_i \rfloor \quad \forall i \tag{28}$$

$$T'' \leq \lfloor (\mathrm{O} + \mathrm{I} + \sum_i S_i)/\sum_i \mathrm{p}_i \rfloor \tag{29}$$

So, being N the number of products in the batch, we will have N + 1 inequalities that limit the value of T'' by constants and again T'' will be defined by the smallest value.

So, the analytical method can be decomposed into 4 steps:

step 1: find T', where:

$$T' = \lfloor \min_{\forall i} \{\mathrm{UD}_i/\mathrm{p}_i\} \rfloor \tag{30}$$

step 2: calculate S_i for all products.

$$S_i = \mathrm{UD}_i - \mathrm{p}_i * T' \quad \forall i \tag{31}$$

step 3: find T'', where:

$$T'' = \lfloor \min\{\min_{\forall i}\{(\mathrm{UO}_i + \mathrm{UI}_i + S_i)/\mathrm{p}_i\}, (\mathrm{O} + \mathrm{I} + \sum_i S_i)/\sum_i \mathrm{p}_i\}\rfloor \tag{32}$$

step 4: calculate T, where:

$$T = T' + T'' \tag{33}$$

$$T = \min\{T, \mathrm{Z}\} \tag{34}$$

It is probably possible to extend this thinking to the solution of a class of linear optimization problems. However, we will work on this concept in an upcoming paper.

Solution for the example presented before

Applying the method for solving the example presented in section 2, we have:

$T' \leq \lfloor 1000 \text{ g}/(60 \text{ g/min})\rfloor$

$T' \leq 16$ min for A

$T' \leq \lfloor 500 \text{ g}/(40 \text{ g/min})\rfloor$

$T' \leq 12$ min for B

so:

$T' = 12$ min

Then:

$S_A = 1000 \text{ g} - 12 \text{ min} * 60 \text{ g/min} = 280 \text{ g}$

$S_B = 500 \text{ g} - 12 \text{ min} * 40 \text{ g/min} = 20 \text{ g}$

And so we have:

$T'' \leq \lfloor(3000 \text{ g} + 600 \text{ g} + 280 \text{ g})/(60 \text{ g/min})\rfloor$

$T'' \leq 64$ min   for A

$T'' \leq \lfloor(2000 \text{ g} + 600 \text{ g} + 20 \text{ g})/(40 \text{ g/min})\rfloor$

$T'' \leq 65$ min   for B

$T'' \leq \lfloor(1000 \text{ g} + 3000 \text{ g} + 300 \text{ g})/(100 \text{ g/min})\rfloor$

$T'' \leq 43$ min   for A and B

So that:

$T'' = 43$ min

So we have:

$T = 55$ min

As $T <$ Z, then $T = 55$ min will certainly be the optimal solution.

In this case we will have:

$P_A = 3300$ g e $P_B = 2200$ g

$D_A = 1000$ g e $D_B = 500$ g

$E_A = 2300$ g e $E_B = 1700$ g

Note that it is not important to know the values of O_i and I_i, $\forall i$, however, after determining a solution, these values can be found by taking a solution from the undetermined system of inequalities:

$$O_i + I_i = E_i \quad \forall i \tag{35}$$

$$O_i \leq \mathrm{UO}_i \quad \forall i \tag{36}$$

$$\sum_i O_i \leq \mathrm{O} \tag{37}$$

$$I_i \leq \mathrm{UI}_i \quad \forall i \tag{38}$$

$$\sum_i I_i \leq \mathrm{I} \tag{39}$$

$$O_i, I_i \in \mathbb{Z}^+ \quad \forall i \tag{40}$$

5. Tests and results

To test the developed model and analytical solution method, we created a solver in C++ (https://github.com/blinded) and a solver in LINGO (https://github.com/blinded). The tests were performed on a notebook with an Intel i7 processor. We tested the solvers developed for the benchmarks presented on Tables 1, 2 and 3, and for randon benchmarks generated by the following functions:

$$\mathrm{p}_i = \mathrm{rand}()\%30 + 10 \tag{41}$$

$$\mathrm{UD}_i = \mathrm{rand}()\%3000 + 800; \tag{42}$$

$$\mathrm{seed1} = \mathrm{rand}()\%3000 + 500; \tag{43}$$

$$\mathrm{seed2} = \mathrm{rand}()\%5000 + 1000; \tag{44}$$

$$\mathrm{O} = \mathrm{N}/2 * \mathrm{seed1}; \tag{45}$$

$$\mathrm{UO}_i = \mathrm{rand}()\%(\mathrm{seed1} - 500) + 500; \tag{46}$$

$$\mathrm{I} = \mathrm{N}/2 * \mathrm{seed2}; \tag{47}$$

$$\mathrm{UI}_i = \mathrm{rand}()\%(\mathrm{seed2} - 1000) + 1000; \tag{48}$$

$$Z = 100; \tag{49}$$

i	1	2	total
p_i	60	40	
UD_i	1000	500	
UO_i	600	600	1000
UI_i	3000	2000	3000
		Z	100

Table 1: Benchmark MBPTMP 001

i	1	2	3	total
p_i	60	40	50	
UD_i	1000	500	800	
UO_i	600	600	600	1500
UI_i	3000	2000	1000	3500
			Z	100

Table 2: Benchmark MBPTMP 002

Randomly generated benchmarks were named RMPBPTMP N, being N the number of products. Seeking to enable the reproduction of the results, in the computational construction of the benchmarks we used the function srand((unsigned) source), where source is a defined value. To build the results presented in this work, we used source=0. The benchmarks used for the tests performed can be consulted at github.com/blinded.

Table 4 presents the results obtained by applying the analytical method (C++ solver) and the LINGO solver for the solution of the previously presented benchmarks.

It is important to point out that the LINGO solver was developed with the purpose of validating the results found by the proposed analytical method.

i	1	2	3	4	5	6	7	8	9	10	total
p_i	60	40	50	40	30	50	60	10	20	40	
UD_i	1000	500	800	500	400	500	2000	300	500	1000	
UO_i	600	600	600	1500	300	200	500	800	0	200	3000
UI_i	3000	2000	1000	800	3000	1000	400	300	200	0	5000
										Z	100

Table 3: Benchmark MBPTMP 003

problem	LINGO solver	time (s)	analytical method	time (s)
MBPTMP 1	55	0.04	55	0.001
MBPTMP 2	48	0.06	48	0.001
MBPTMP 3	30	0.09	30	0.001
RMBPTMP 20	100	0.08	100	0.001
RMBPTMP 50	98	0.12	98	0.001
RMBPTMP 100	98	0.12	98	0.002
RMBPTMP 1,000	78	3.10	78	0.002
RMBPTMP 10,000	70	168.87	70	0.006

Table 4: Results obtained with the LINGO solver and the analytical method

As the analytical method is a polynomial time complexity method, we did not intend to compare computational costs, however it is possible to verify that both solvers are capable of finding optimal solutions for the proposed benchmarks very quickly, even for very large problems.

We have considered here a one-day period problem, so in a future work we will study the complexities of solving a multi-period problem and try to propose new solution methods if needed.

6. Contributions to this work

The work for the elaboration of this paper was carried out by tree contributors, as described below. Blinded, developed the detailed study on the plastic bag production process, which made it possible to identify the problem addressed in this paper. Blinded was responsible for identifying the problem, mathematical modeling, proposing an analytical solution, developing a solver in C ++ for applying the analytical solution and a solver in LINGO for validating the mathematical model and the results found by the first solver, developing benchmarks, and testing the model and analytical method proposed, as well as surveying bibliography and elaboration of the paper. Blinded contributed to the bibliographic survey. The authorship of this work is distributed according to contributions described.

7. Conclusions and suggestions for future works

In this paper we presented the Multi-product Batch Processing Time Maximization problem, as well as a mathematical model and an analytical solution method for this problem. The mathematical model and analytical method were tested, respectively, by a solver developed with the LINGO software, from LINDO Systems, and with a solver developed in C++ language. Optimum results were found for all proposed benchmarks, very quickly, even in the case of very large problems, which demonstrates the efficiency of the proposed method. As in this paper we considered a planning period of one day, in a future work we will study the complexities that arise when considering the same problem in a multi-period scenario. We will also verify if there is the possibility of extending the proposed analytical method to solve a class of linear integer programming problems with similar characteristics to the studied problem.

8. Acknowledgments

Blinded.

References

Eilon. (1985). Multi-product batch production on a single machine - A problem revisited. *OMEGA Int. J. of Mgmt Sci.*, Vol. 13 (5), pp. 453–468.

Kim, M., Jung, J. H. and Lee, I. (1996). Intelligent scheduling and monitoring for multi-product networked batch processes. *Computers chem. Engn*, Vol. 20 (Suppl.), pp. 1149–1154.

Liu, G., Li, F., Yang, X., and Qiu. S. (2020). The multi-stage multi-product batch-sizing problem in the steel industry. *Applied Mathematics and Computation*, Vol. 369, 124830.

Méndez, C.A., Henning, G.P., Cerdá, J. (2000). Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. *Computers and Chemical Engineering*, Vol. 24, pp. 2223–2245.

OMEGA Journal. (1993). Single Machine Multi-product Batch Scheduling: Testing Several Solution Methods. *OMEGA Int. J. of Mgmt Sci.*, Vol. 21 (6), pp. 709–711.

Ravemark, D. E., and Rippin, D. W. T. (1998). Optimal design of a multi-product batch plant. *Computers chem. Engng*, Vol. 22 (1-2), pp. 177–183.

Shi, B., Qian, X., Sun, S., Yan, L. (2017). Rule-based scheduling of multi-stage multi-product batch plants with parallel units. *Chinese Journal of Chemical Engineering*, in press.