INTEGRATED APPROACH FOR THE CAPACITATED LOT SIZING AND NO-WAIT FLOWSHOP SCHEDULING PLANNING PROBLEM

Bernardo V. Furtado[1], Martin G. Ravetti[1*] and Mauricio C. de Souza[1*]

*Departamento de Engenharia de produção - Universidade Federal de Minas Gerais
[1]Avenida Presidente Antonio Carlos, 6627, Pampulha - Belo Horizonte/MG
E-mails: bernardovf@ufmg.br / martin@dcc.ufmg.br / mauricio@dep.ufmg.br /

ABSTRACT

This article deals with the lot sizing and scheduling problem in a one-level manufacturing based on a real case. It is proposed a two-level integration scheme, where a MILP model for the lot sizing problem is optimally solved in the comercial software CPLEX. As for the scheduling subproblem, three approaches are proposed and compared with different objetive functions. The shop floor is composed of no-wait flowshop with family-setup times. The master production model is performed once for the time horizon, and three heuristics are executed in accordance to the outputs of the lot-sizing once per time period. It is tested up to 150 instances and the makespan objetive leads to a better solution for the problem.
Keywords: Production planning. Lot sizing. Machine scheduling
AD & GP, IND

RESUMO

É considerado no artigo o problema de dimensionamento de lotes capacitado e o problema de sequenciamento de máquinas em um único nível de manufatura. É considerado um caso real. É proposto um método de duas fases, onde o modelo PLIM para o problema de dimensionamento de lotes é resolvido de maneira ótima no software comercial CPLEX . Para o problema de sequenciamento são propostas e comparadas três heurísticas com diferentes funções objetivo. O chão de fábrica é composto por *no-wait flowshop* com tempos de setup por famílias de produtos. O problema de dimensionamento de lotes capacitado é resolvido uma única vez para o horizonte de tempo enquanto as três heurísticas são executadas de acordo com as saídas do dimensionamento de lotes uma vez por instante de tempo. São testadas mais de 150 instancias e foi concluido que o objetivo de *makespan* conduz a melhores resultados para o problema proposto.
Palavras chave: Planejamento da produção. Dimensionamento de lotes. Sequenciamento de máquinas
AD & GP, IND

1 Introduction

In manufacturing, the role of production planning is to define how many and when production batches will be produced. The main concern of production planning decisions is the trade-off between cost and customer satisfaction. Costs are generally defined of inventory and production costs, while customer satisfaction can be measured by quality and lead time.

The master program or capacitated lot sizing problem acts in the medium term of an organization. Its main function is to take manufacturing decisions, when receiving as input the independent-demand forecast and shop-floor's capacity estimation, taking into account inventory, production and setup costs. The lot-sizing models have various classifications and variations, for more details see Karimi *et al*(2003), Pinedo(2005) and Pochet and Wolsey(2006). The Capacitated lot sizing and scheduling problem (CLSP) is NP-hard, we refer the reader to Bitran(1982) for a review on lot sizing models complexity

Soon after the lot-sizing decisions are made, production lots should be decomposed into production orders to run on machines (jobs), and that is when the operational planning takes place by dealing with a scheduling problem. The scheduling problem is defined as determining the order of a set of jobs on the available machines aiming at certain optimization criteria, usually time related.

The problem here considered is integrate these two levels and it is based on a real case. Were a novel iterative method of integration between the lot-sizing and scheduling problems is proposed and analized. The programming model with a commercial solver (CPLEX) is used to optimally solve the lot-sizing problem, while for each time interval (period) the related scheduling problem is heuristicaly solved.

The company in analysis produces steel tubes. Its production process consists on seven stages: oven, perforator mill, PQF mill, reheating oven, finishing mill, quality tests and cut-off press. The second and third stages are the bottleneck stages, so they are considered critical. These two mills are basically two machines in sequence, characterizing a flowshop environment. There is no intermediate buffer between the two mills because the tubes can not chill during the process, thus it can be called a no-wait flowshop system.

Many studies and articles have been published discussing integration methods between lot sizing and scheduling planning problems. Nevertheless, most of the studies deal with problems of single machine, in which only the setup constraint is modified, and there are no many references regarding complex shop floor environments. Drexl and Kimms(2000) published a survey of various methods to integrate the two problems according to the setup treatment and time horizon. Following a similar approach we can cite Pochet and Wolsey(2006) and Karimi *et al.*(2003). On parallel machines there are also a few works. Mateus *et al.*(2010) propose an iterative method to the unrelated parallel machines problem: the lot-sizing problem in this case is solved optimally while the scheduling problem is solved by a GRASP heuristic. We follow their approach modifying the integration procedure.

The article is organized as follows: In section 2 we propose a mixed integer-linear programming model for capacitated lot sizing, and presented three heuristics for the no-wait flowshop problem with different optimization objectives. In Section 3 we propose the method of integration between the two problems. Computational results are presented in Section 4, conclusion in Section 5 and comments in Section 6.

2 Medium-term and short-term levels

In the lot-sizing problems, the basic input is the aggregate production capacity and external demand. As output we have the quantity of each product that must be produced by period considering as criterion optimization costs (setup, production and stock). In each individual period we have the short-term problems, that is, to schedule the production orders in a way that the products' dead lines and the shop capacity limits are satisfied. Here the main concern is to deliver the production orders (jobs) before the delivery dates (due dates). The optimization criteria is time-related, such as minimizing total delay, weighted delays, total completion time, maximum delay etc.

2.1 Capacitated lot sizing

According to the classification of production planning models given by Drexl and Kimms(2000), Karimi *et al.*(2003) and Pochet and Wolsey(2006) the following model can be had as big bucket, in which various products can be produced by each period, and scheduling decisions are not taken at this level. Furthermore the model is multi-item and considers only one level of production. As for not meeting demand, backlogging is permitted provided that a fine is paid.

A market research or external requests generates a demand of product p, $p = 1,\ldots,P$ in period t, $t = 1,\ldots,T$.

The parameters are: a_p is the amount of available capacity consumed by production of product p [time units], b_p is the amount of available capacity consumed by setup of product p [time units], d_{pt} external demand of product p in period t [units]., w_t machine available capacity in period t [time units].c_p, h_p, v_p and q_p are the costs (production, inventory, backlogging and setup, respectively)

The decision variables are: quantity of product p produced in t (x_{pt}), quantity of p left in inventory in t (s_{pt}) and quantity of product p backlogged for t+1(n_{pt}). y_{pt} is the binary variable that indicates if a setup of product p is run in t (y_{pt}=*1*) and y_{pt}=0, else.

The model is descibed below,

$$\text{Minimize} \sum_{t=1}^{T}\sum_{p=1}^{P} c_p x_{pt} + h_p s_{pt} + v_p n_{pt} + q_p y_{pt}$$

$$s_{p(t-1)} - s_{pt} + x_{pt} + n_{pt} - n_{p(t-1)} = d_{pt} \qquad \forall t \in T \quad \forall p \in P \qquad (1)$$

$$\sum_{p=1}^{P} a_p x_{pt} + b_p y_{pt} \le w_t \qquad \forall t \in T \qquad (2)$$

$$a_p x_{pt} \le w_t y_{pt} \qquad \forall t \in T \quad \forall p \in P \qquad (3)$$

$$y_{pt} \in \{0,1\} \qquad \forall t \in T \quad \forall p \in P \qquad (4)$$

$$x_{pt}, s_{pt}, n_{pt} \in Z_+ \qquad \forall t \in T \quad \forall p \in P \qquad (5)$$

The objective function is minimizing the sum of setup, production, inventory and backlogging costs. Constraints (1) represent the demand balance. The second group, (2), ensure that everything that is produced does not exceed the available capacity in the period. Constraints (3) indicate that in each period that product p is produced it is necessary running and paying a setup. The setup variables are binaries (4) and inventory, production and backlogging are integers and non negative (5).

2.2 No-wait flowshop scheduling problem

As mentioned, the shop floor scenario considered in this paper is a flowshop, where N jobs must be sequenced in two unrelated machines. Jobs are sorted into families or classes in which they have similar production characteristics. Among jobs of the same class there are no need to have a setup between them. For a deeper understanding about setup and families of products see Allahverdi *et al.*(1999) and Pinedo(2005). The no-wait constraint means that there should be no time intermediate buffer between two operations of a job. Problems of this nature, well known to be NP-hard, receive considerable attention, where heuristic methods are the more common approaches. Although no-wait flowshop has received only a few attempts of study, it has a large applicability in industrial problems.

This problem consists on a set of N jobs to be scheduled in two machines. Let job $j[i, j]$ be the j*th* job in the i*th* batch. $p1_{[ij]}$ and $p2_{[ij]}$ denote the processing times of job [i,j] on machines 1 and 2, respectively. $s_{[i],k}$ is the setup time of the i*th* batch on machine k. β_i is the number of jobs in i*th* batch in the schedule.

Three approaches are compared for this scenario: The first subproblem(SP1) (According to Pinedo(2005), $F2|nwt, fmls|L_{max}$) is proposed to solve the objective function of minimizing the maximum lateness, which was proposed by Cheng and Wang(2006). This first method has computational complexity $O(n^2 logn)$ and reaches about 5% GAP in average for real-sized instances.

It combines an inicial solution, obtained from EDD (early due dates) algorithm with a merging batches method.

Second subproblem (SP2) (According to Pinedo(2005), $F2|nwt, fmls|\sum T_i w_i$), considered in this article objective to minimize weighted tardiness. Considering the objective function of minimizing weighted tardiness is important when some jobs are more important than others and in some cases the weight can be considered cost of tardiness, and minimize the sum between cost (weight) and tardiness can be financially interesting. The second heuristic is a dispaching rule, which is $O(nlogn)$. In this case jobs are scheduled in increasing order of the index below:

$$I(i) = \frac{d_i}{w_i} \tag{6}$$

d_i is the due date of job i, and w_i is the weight of i, defined by its backlogging cost.

Third subproblem (SP3) (According to Pinedo(2005), $F2|nwt, fmls|C_{max}$), objective to minimize makespan. When family setup constraint is considered the most efficienty method to minimize makespan is dividing the schedule into batches. Into these batches setup is not needed, so the flowtime in system is minimized. The heuristic consists on two stages. Firstly it obtains as inicial solution by joining the jobs of the same classes and then permutate these batches and find the best solution.

To improve the results were applied a deterministic local search heuristic based in the insertion algorithm which was proposed by Nawaz *et al.*(1983). Its complexity is $O(n^2)$. Several experiments were done and this local search improve the solution nearly 3%.

We must remember that our goal is to minimize the cost of production planning. Our methodology is comparing the three objetives functions to test which one fits better the objetive.

3 Integration algorithm

It is proposed in this paper an interative algorithm for integration, the algorithm consistis in two modules: firstly, the CLSP model is run optimally by using the commercial solver CPLEX, and the results are the inputs of scheduling problem. The production lots are decomposed into jobs. Scheduling module determines the feasibility of lot sizing output. If the schedule is not feasible, jobs have to be delayed to the next subperiod, these jobs are considered critical and their due dates for the next period are negative to ensure that they will be sequenced firstly. If the schedule is feasible and there is idle capacity, the idle capacity is used to attend next period of scheduling. The integration scheme is shown in Figure **??**:

It is considered in the algorithm the following variables: α_{pt} is the quantity of product p that has to be delayed to the next period and ϕ_t is the idle capacity in period t.C_l and C_s are the lot sizing and scheduling costs, respectively. Denote N_c the number of jobs of class c, ans I_t the total numeber of jobs in period t.

The procedure of decomposing lots into jobs consists on disaggregate a_p and b_p variables into $p1, p2, s1, s2$. a_p is the sum between processing times and b_p is the sum between setup times.

The capacity considered in lot sizing is an aggregate one thus it does not consider the shop floor specificities. No-wait flowshop environment has a particularity: there are superposition of jobs the schedule. Thus, the capacity constraints of lot sizing must be adjusted by multiplying a factor, called ξ.

$$\sum_{p=1}^{P} a_p x_{pt} + b_p y_{pt} \leq \xi w_t \tag{7}$$

$$a_p x_{pt} \leq \xi w_t y_{pt} \tag{8}$$

$$\xi \geq 1 \tag{9}$$

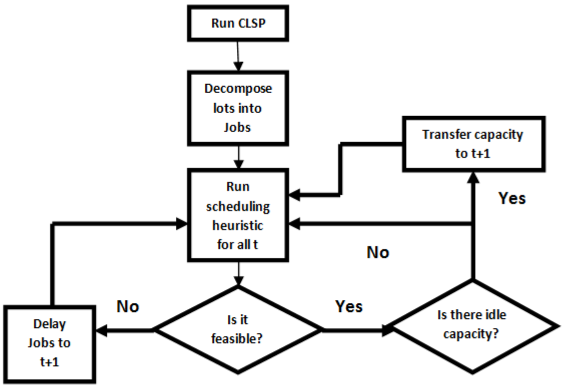

Figure 1: *Integration scheme: lot sizing is run once and scheduling heuristics are run for all time horizon*

Algorithm 1 Integration algorithm

Run Capacitated lot sizing model
$C_l \leftarrow lotsizingcost$
$C_s \leftarrow 0$
for $t = 1$ to $t = T$ **do**
  Decompose lots into jobs
  Run scheduling heuristic for subproblems (SP1, SP2 or SP3)
  for $i = 1$ to $i = I_t$ **do**
    if $C_i > w_t$ **then**
      $\{\alpha_{pt}\} \leftarrow \{\alpha_{pt}\} + i$
      $d_i \leftarrow -(C_i)$
      $C_s \leftarrow C_s + v_p$
    end if
  end for
  $\phi_t \leftarrow max\{w_t - C_{max}, 0\}$
  $w_{t+1} \leftarrow w_{t+1} + \phi_t$
end for
$C_l \leftarrow C_l + C_s$

ξ value is determined empirically. It was created a set of independent instances considering various scenarios, diferent from the originals, to determine this value. Consider two situations: S1 is the sum of processing times and setups of the two machines and S2 is the regular no-wait flowshop situation. ξ is calculated in equation 10:

$$\xi = \frac{C_{max}(S1)}{C_{max}(S2)} \tag{10}$$

It is trivial to see that situation s(S1) is an upper bound of situation 1(S1). The upper bound is defined as:

$$C_{max}(S1) = \sum_{j=1}^{J} \sum_{i=1}^{\beta_j} p1_{[i,j]} + p2_{[i,j]} + s_{[i],1} + s_{[i],2} \tag{11}$$

The subproblem used was subproblem 3(SP3) with makespan objective, which leads to the best results. ξ value follows a normal distribution ($p - Value = 0,548$ for Anderson-Darling normality test). The 95% confidence interval of ξ is: $(1,55258 \leq \xi \leq 1,58042)$. Thus, it is arbitrarily adopted that $\xi \approx 1,6$

4 Computacional results

Several numerical experiments were designed and run to test the efficiency of the heuristics and the integration method. The intention here is to compare the efficiency of the three proposed heuristics in some scenarios considering costs.

4.1 Instances generation

The instances were generated by the pseudo-random number generator *Mersenne Twister algorithm*, (see Matsumoto and Nishimura(1998)). The planning horizon has three periods ($T = 3$) for all instances. The number of products varies between 5,10,15 and 20. It was tested four scenarios

These values is based on a real case, where there are approximately 20 products and 200 jobs. The instances distribution and the scenarios utilized are shown in table **??**:

Table 1: *Instances generation: Values were generated following a uniform distribution*

Input Data	Value(Scenario 1)	Value(Scenario 2)	Value(Scenario 3)	Value(Scenario 4)
a_p	U(7,10)	U(7,10)	U(7,10)	U(7,10)
b_p	U(1,2)	U(1,2)	U(1,2)	U(1,2)
d_{pt}	U(1,50)	U(1,50)	U(1,25)	U(1,25)
c_p	U[1,2]	U[1,2]	U[1,2]	U[1,2]
h_p	U[5,10]	U[5,10]	U[5,10]	U[5,10]
v_p	U[40,80]	U[40,80]	U[40,80]	U[40,80]
q_p	U[2,4]	U[2,4]	U[2,4]	U[2,4]
w_t	600	1200	600	1200

4.2 First experiment

The first experiment shows the sensibility of ξ value in the objective function. ξ varies from $1,0$ to $2,0$ and costs for these values are shown in Figure **??**. For values greater than $2,0$ costs are constant.

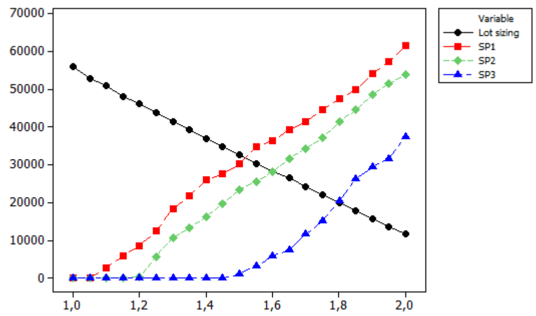

Figure 2: *Lot sizing and three heuristic costs according to ξ increasing. x axis represents ξ and y axis represents cost.*

It is already clear that SP3 perfoms better results for any ξ value. It can be seen that the integration method is worse according to the increasing of ξ value. In practice, ξ can not be less than $1,55$.

4.3 Second experiment

The second experiment relate the scenarios and the number of products to compare the heuristics performance. The number of products varies between 5,10,15 and 20 products. The results are shown in tables **??** and **??**

Table 2: *Algorithm Results: LS represents lot sizing costs. Total is the sum of lot sizing and scheduling costs (average ± standard deviation).*

Nprod	Scenario	LS	Total(SP1)	Total(SP2)	Total(SP3)
5	1	$3930,8 \pm 2695,84$	$9859 \pm 5488,23$	$9099,5 \pm 4878,53$	$6531,4 \pm 3445,56$
	2	$278,3 \pm 93,97$	$278,3 \pm 93,97$	$278,3 \pm 93,97$	$278,3 \pm 93,97$
	3	$230,3 \pm 30,74$	$230,3 \pm 30,74$	$230,3 \pm 30,74$	$230,3 \pm 30,74$
	4	$253,3 \pm 24,81$	$253,3 \pm 24,81$	$253,3 \pm 24,81$	$253,3 \pm 24,81$
10	1	$41475,2 \pm 8395,62$	$48360,6 \pm 10299,6$	$47691,6 \pm 10054,54$	$44556,8 \pm 8682,33$
	2	$8394,1 \pm 7538,83$	$18367,4 \pm 12328,09$	$16970,4 \pm 12148,66$	$12159,9 \pm 9711,91$
	3	$4465,9 \pm 3156,52$	$11495,1 \pm 4349,53$	$10263,8 \pm 3969,77$	$7337 \pm 3403,86$
	4	$476,2 \pm 53,61$	$476,2 \pm 53,61$	$476,2 \pm 53,61$	$476,2 \pm 53,61$
15	1	$84482,3 \pm 17302,81$	$91792,9 \pm 19057$	$91841,3 \pm 17490,2$	$87736,7 \pm 17357,9$
	2	$53178,2 \pm 12005,20$	$68408,8 \pm 12743,8$	$66682,3 \pm 12420,2$	$60058,8 \pm 12175,3$
	3	$25602,3 \pm 5264,15$	$32912,9 \pm 5941,2$	$32961,3 \pm 5788,0$	$28856,7 \pm 5196,2$
	4	$1259,7 \pm 732,38$	$5057 \pm 4769,6$	$4236,9 \pm 3977,7$	$3412,9 \pm 2586,4$
20	1	$128849 \pm 14792,41$	$139919,6 \pm 15972,0$	$136602,4 \pm 14861,4$	$132282 \pm 14878,9$
	2	$83343,6 \pm 16697,9$	$98574,2 \pm 17631,1$	$96835,8 \pm 17303,4$	$90183,2 \pm 17015,2$
	3	$44370,8 \pm 6924,6$	$55441,4 \pm 7833,2$	$51821,7 \pm 6824,3$	$47646,1 \pm 6878,7$
	4	$12400,5 \pm 3979,9$	$27931,1 \pm 3917,6$	$26115,5 \pm 4091,7$	$19110,6 \pm 3917,4$

Table 3: *Algorithm Results: Relative percentage deviation (RPD) is defined as the deviation of the total cost from lot-sizing cost (lower bound) (average). Values in boldface are the best for each scenario*

Nprod	Scenario	RPD(SP1) %	RPD(SP2) %	RPD(SP3) %
5	1	$180,45$	$162,54$	**93,76**
	2	0	0	**0**
	3	0	0	**0**
	4	0	0	**0**
10	1	$16,42$	$14,89$	**7,61**
	2	$193,38$	$140,70$	**57,66**
	3	$231,38$	$196,56$	**102,19**
	4	0	0	**0**
15	1	9,71	9,04	**4,05**
	2	29,74	26,42	**13,49**
	3	29,53	29,39	**13,19**
	4	361,19	281,81	**167,25**
20	1	8,60	6,08	**2,68**
	2	18,90	16,77	**8,52**
	3	25,26	17,22	**7,55**
	4	142,27	125,59	**62,40**
Average	1,2,3,4	77,90	64,01	**33,77**

It is clear that SP3 performs better for all instances because dealing with makespan objective reduces a lot the quantity of jobs delayed and the other subproblems proposed are due date related.

Another important analysis is about scenarios: Scenario 1 is the most tight one because it has high demand and low capacity. It can be seen in Table 2 and Table 3 that scenario 1 leads to the highest cost while scenario 4 leads to the lowest cost.

5 Conclusions and Future research

In this paper we propose an iterative scheme to integrate lot sizing and scheduling problems. The method alternate two decision levels. The CLSP is solved optimally and scheduling is solved by three diferent heuristics with diferent objetive functions. Different approaches are compared in computational results. It can be concluded that with the integration scheme proposed makespan objetive leads to the lowest cost because backlogging in scheduling is defined as which jobs are finished after the time capacity (in hours). Minimizing the completion time of the last job consequently minimizes the quantity of jobs delayed.

We solved instances with 25 produtcs and up to 300 jobs, it is a real-sized problem from a real organization.

Future works may include a detailed analysis considering new forms of integration as well as a no-wait flowshop mixed-integer linear program. A direction of research is to consider that the idle capacity returns to lot sizing model, and CLSP is run again.

6 Comments

The student has implemented the three approaches analysed in C++ language, MILP model in CPLEX solver and has written the article. This article is under submission for INFORMS and will be submitted to a journal. This project was financed by FAPEMIG since March/2011.

References

Allahverdi, A., Gupta, N.D.J and Aldowaisan, T. (1999), A review of scheduling research involving setup considerations. *Omega, Int. J. Mgmt Sci*, 27: 219–239.

Bitran, G. and Yanasee, H. (1982), Computational complexity of capacitated lot sizing problem. *Management Science*,28.

Drexl, A and Kimms, A. (2000), Lot-sizing and scheduling: survey and extensions. *European Journal of Operations Research*,99: 221–235.

Karimi, B. Ghomi,F and Wilson, M. (2003), The capacitated lot sizing problem: a review of models and algorithms. *Omega*,31: 378–365.

Pinedo, M. (2005), Scheduling: theory, algorithms and systems. *New York: Prentice Hall. Berlin, Springer*

Pochet, Y and Wolsey, L.A. (2006), Production planning by mixed integer problem. *Berlin, Springer*

Mateus, G.,Ravetti, M. , Souza, M. and Valeriano, T. (2010), Capacitated lot sizing and sequence dependent setup scheduling: an iterative approach for integration. *Journal of Scheduling*,13: 245–259.

Matsumoto, M. and Nishimura, T. (1998), Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*,8: 3–30.

Nawaz, M., Enscore, E. and Ham, I. (1983), A heuristic algorithm for the m-machine, n-job flowshop sequencing problem. *Omega*, 11: 91–95.

Ruiz, Wang. W. and Cheng T.C.E. (2006), A heuristic approach for two-machine no-wait flowshop scheduling with due dates and class setups. *Computer & Operations Research*,33: 1326–1344.

.