Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

The Generalized Single-Facility Capacitated Lot-Sizing and Scheduling Problems

Fayez F. Boctor

August 2016

CIRRELT-2016-43

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2016-011.

The Generalized Single-Facility Capacitated Lot-Sizing and Scheduling Problems

Fayez F. Boctor*

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, Université Laval, Pavillon Palasis-Prince, 2325, de la Terrasse, Québec, Canada G1V 0A6

Abstract. This paper deals with a new and more realistic version of the lot sizing and scheduling problem where we have a single facility that process a number of different products and for each product we have a set of delivery dates and the quantity to deliver at each of these dates. The objective is to minimize the sum of setup costs and inventory holding costs. Setups can be carried over between two successive production lots if the same product is to be processed. A mathematical formulation of this problem is given as well as two new solution heuristics. As the literature does not provide any other method to solve this problem, only the results obtained by these two heuristics are compared.

Keywords. Lot-sizing and scheduling, heuristics, integer programming.

Acknowledgements. This research work was partially supported by Grant OPG0036509 from the National Science and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.

* Corresponding author: Fayez.Boctor@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

1- INTRODUCTION:

The standard *Capacitated Single-facility Lot-sizing Problem* (CSLP) assumes that the planning horizon is divided into a number of time intervals (or time periods) and makes three implicit and simplifying assumptions. First, it implicitly assumes that a lot that starts within given time period should be finished within this same period. This assumption is needed to simplify the mathematical expression of the capacity constraints. Second, it assumes that setups cannot be carried over from one period to the next. Thus if the last run of a period and the first of the next one process the same product, a setup cost for each of these two runs are included in the objective function. This overestimates the overall setup cost as, in most cases, these two lots can be processed with one setup (see Jans et al 2008).

Third, although it is assumed that delivery occurs only at the end (or the beginning) of each time period, the objective function does not include the inventory holding cost between the time a lot is finished and the beginning of the next period. This is also assumed to simplify the mathematical expression of the objective function and to make the determination of processing dates unnecessary. However this under estimate the real inventory holding cost.

Some research publications (see Sox et al 1999) propose to relax the first assumption and allow setup carryover. However the resulting formulation still does not include the inventory holding cost between the time a lot is finished and the beginning of the next period.

A more realistic version of the problem, called hereafter the *Generalized Single-Facility Capacitated Lot-Sizing and Scheduling Problem* (GSCLSP), is studied in this paper. In this version each product has a set of delivery dates and the quantity to deliver at each of these dates is known. It is allowed to start the production of a lot at any time and finish it at any time later within the finite planning horizon provided that the required demands are delivered at the required dates. Finally, the objective function of this new formulation includes the inventory holding cost of each produced lot from the end of its processing until the delivery of all its units.

Assuming that all delivery dates (of all products) are arranged in their ascending order, denoted, $t_1,t_2,..,t_L$, we may consider that the planning horizon is divided into L time intervals of unequal lengths, where the length of interval l is the time interval between t_{l-1} and t_l. We also allow that a lot that starts within a given time interval, say interval l, can be finished within the same interval or within a later interval, say interval $l+r$.

2- LITERATURE REVIEW

A usual classification of deterministic capacitated lot sizing problems is based on four main criteria: the nature of the demand (static or dynamic), the number of products to process (single or multiple), the number of production facilities involved (one or more than one) and the number of production stages (single or multiple). In the case of static demand, single production stage, single machine and multiple products to process, the problem is called the *Economic lot sizing problem* (ELSP). Earlier solution methods for the ELSP focus on cyclic schedules which satisfy the Zero-Switch-Rule (ZSR), meaning that an item is produced only if its inventory depletes to a zero level and lots of the same product are of the same size. Elmaghraby (1978) presents a review and some extensions of such methods and Boctor (1987) presents a review of some of some other solution procedures. Delporte and Thomas (1978) and Dobson (1987) presented solution approaches that produce time-varying lot size solutions and do not require equal production lots. Raza et al (2008) provide a comparison of some of these time-varying lot sizing solutions.

In this paper we are rather concerned with the lot sizing problem with dynamic demand. The standard version of this problem is named the capacitated lot sizing problem (CLSP).A review of the single product version of this problem is given in Brahimi et al (2006). An extensive review of the different versions of the multi-product problem is presented in Jans et al (2008) and in Gicquel et al (2008). The multi-stage multi-product problem is reviewed in Buschkuhl et al (2010).

Few publications allow for setup carryover from period to period. Gopalakrishnan et al (1995) and Gopalakrishnan (2000) present a formulation of the CLSP with setup carryover. Their formulation assumes that all products have the same setup cost and time, and uses twice as many binary variables as the formulations given in Sox et al (1999). Gopalakrishnan et al (2001) provide a Tabu search method to solve the problem. Porkka et al (2003) showed that allowing for setup carryover can produce substantial savings.

Although the literature dealing with lot sizing problems is rich, to the best of my knowledge there is no publication that deals with the *Generalized Single-Facility Capacitated Lot Sizing and Scheduling Problem* (GSCLSP) introduced in this paper.

3- MATHEMATICAL FORMULATIONS

In this section, a mathematical formulation of the Generalized Single-facility Capacitated Lot Sizing and Scheduling Problem is presented after presenting the used assumptions and notation.

This formulation allows for setup carryover, takes into account the inventory holding cost for each produced lot between the finish of its processing and the delivery of all its units. Also, it determines the sequence and processing dates of all lots to process.

Assumptions:

1- A number of products are to be produced by a single production facility. Each lot is composed of a number of units of the same product and the production facility cannot processes more than one lot at a time;
2- A finite planning horizon;
3- For each product there is an upper limit on its lot size;
4- Processing time of a lot of a given product is composed of the processing time of its units plus a known sequence-independent setup time. Processing time of the units composing a lot of a given product can be either proportional to the quantity to produce or constant. In this later case, lot processing time depends only on the product to process and is independent from the quantity to process. Both cases will be modeled hereafter. However the test instances used in the computational experiment reported in section 6 use constant processing times;
5- Lots of the same product are not necessarily of same size and not necessarily uniformly distributed aver the planning horizon;
6- Delivery dates and quantities to deliver at these dates are known and deterministic;
7- No backlogging is allowed;
8- Two cost elements are considered: setup cost and inventory holding cost;
9- For each product, unit inventory holding cost per time unit as well as setup cost are constant.

Notation:

Parameters

N	number of different products to produce; indexed i
T	set of delivery dates indexed in the ascending order; $T=\{t_l;\ l=1,.., L\}$
d_{il}	quantity of product i to deliver at t_l. This demand is nil if it is not required to deliver any quantity of product i at t_l
p_i	processing time of a unit of product i
s_i	setup time for a lot of product i
c_i	setup cost of a lot of product i
h_i	inventory holding cost of a unit of product i per time unit

Q_i upper limit on the size of a lot of product i

F_i the required inventory level of product i at the end of the planning horizon

Décision variables

x_{ipl} a binary that takes the value 1 if a lot of product i is in position p among those to finish between t_{l-1} and t_l (even if it starts before t_{l-1}). Notice that, as we have an upper limit Q_i on the lot size of i, more than one lot of product i may be processed and finished between t_{l-1} and t_l but in different positions in the sequence

q_{ipl} the quantity of product i if produced in the p^{th} position and finishes between t_{l-1} and t_l

f_{ipl} the finish date of product i if produced in the p^{th} position and finishes between t_{l-1} and t_l

I_{il} inventory level of product i at t_l just after delivering the demand d_{il}

Using this notation, a formulation of the GSCLSP is given. First, the case where lot processing time depends on the quantity to process is presented (Model 1) and then the case of constant processing times is modeled (Model 2).

***MODEL* 1: Lot processing time depends on the quantity to process**

Find $x_{ipl} \in \{0,1\}$, $q_{ipl} \geq 0$, $f_{ipl} \geq 0$, and $I_{il} \geq 0$; $i=1,..,N$; $p=1,..,N$; $l=1,..,L$, which:

$$Minimize: \sum_{i=1}^{N}\sum_{l=1}^{L} h_i I_{il-1}\,(t_l - t_{l-1}) + \sum_{i=1}^{N}\sum_{p=1}^{N}\sum_{l=1}^{L} h_i q_{ipl}\,(t_l - f_{ipl}) + \sum_{i=1}^{N}\sum_{p=1}^{N}\sum_{l=1}^{L} c_i x_{ipl} \quad (1)$$

$$Subject\ to: \sum_{i=1}^{N} x_{ipl} \leq 1 \quad ;\ p = 1,..,N, l = 1,\dots,L \quad (2)$$

$$\sum_{i=1}^{N} x_{ipl} \leq \sum_{i=1}^{N} x_{i,p-1,l} \quad ;\ p = 2,..,N, l = 1,\dots,L \quad (3)$$

$$q_{ipl} \leq x_{ipl} Q_i \quad ; i = 1,\dots,N;\ p = 1,..,N, l = 1,\dots,L \quad (4)$$

$$I_{il} = I_{i,l-1} + \sum_{p=1}^{N} q_{ipl} - d_{il} \ ; i = 1,\dots,N;\ l = 1,\dots,L-1 \quad (5)$$

$$I_{i,L-1} + \sum_{p=1}^{N} q_{ipL} - d_{iL} \ = F_i \ ; i = 1,\dots,N \quad (6)$$

$$f_{i11} \geq x_{i11}(s_i + p_i q_{i11}) \ ; i = 1,\dots,N \quad (7)$$

$$f_{i1l} \geq x_{i1l} t_{l-1} \quad ; i = 1,\dots,N;\ l = 2,\dots,L \quad (8)$$

$$f_{ipl} \geq x_{ipl} f_{j,p-1,l} + s_i x_{ipl} x_{j,p-1,l} + p_i x_{ipl} q_{ip,l} \quad ; i = 1,\dots,N; j = 1,..,N; p = 2,..,N;\ l = 1,\dots,L \quad (9)$$

$$\sum_{i=1}^{N}\sum_{p=2}^{N} s_i x_{ipl} + p_i q_{ipl} \ \leq t_l - \sum_{i=1}^{N} x_{i1l} f_{i1l} \ ; l = 1, \dots, L \tag{10}$$

The first term in the objective function (1) gives the inventory holding cost of items over the time intervals t_{l-1} and t_l. The second term gives the inventory holding cost of the produced items between the end of their processing and the following delivery date. The third term gives the setup cost of the processed lots. Constraints (2) require that there is at most one product in each position of each time interval (i.e., the interval between two consecutive delivery dates). Constraints (3) make sure that if there is no product in a position then there are no products in the next position. Constraints (4) make sure that the produced quantities do not exceed the lot size upper limit. Constraints (5) and (6) determine the inventory levels and assure that the demands are fulfilled without backlogs. Constraints (7), (8) and (9) determines the finish times of the lots to produce. Finally, constraints (10) make sure that we have enough time to produce the required quantities in each time interval (capacity constraints).

This model is difficult to solve as it contains a large number of variables and constraints. It is composed of N^2L binary variables, $NL(2N+1)+L$ continuous variables and $NL(2N+1)+L$ constraints. Thus if we have 10 products and 20 delivery dates our model has 2000 binary variables, 4220 continuous variables and 4220 constraints. It is also important to notice that the objective functions (1) as well as constraints (9) and (10) are quadratic which adds to the difficulty of solving the model.

***MODEL* 2: Constant lot processing times**

In some industrial settings, for example in some chemical production facilities, lot processing time of a product *i* (including its set up time), denoted P_i, can be independent of the quantity to be processed. In this case the above presented model can be modified as follows. Replace constraints (7), (9) and (10) respectively by:

$$f_{i11} \geq x_{i11} P_i \ ; i = 1, \dots, N \tag{11}$$

$$f_{ipl} \geq x_{ipl} f_{j,p-1,l} + P_i x_{ipl} x_{j,p-1,l} \ ; i = 1, \dots, N; j = 1, \dots, N; p = 2, \dots, N; \ l = 1, \dots, L \tag{12}$$

$$\sum_{i=1}^{N}\sum_{p=2}^{N} P_i x_{ipj} \leq t_l - \sum_{i=1}^{N} x_{i1l} f_{i1l} \ ; l = 1, \dots, L \tag{13}$$

Let us present now some feasibility conditions that can be used a priori to check whether a feasible solution exists or not. These conditions are useful to generate test instances with feasible solutions. A necessary but not sufficient condition for having a feasible solution for Model 1 is:

$$\sum_{i=1}^{N}(s_i\left\lceil\frac{\sum_{j=1}^{l} d_{il}}{Q_i}\right\rceil + \sum_{j=1}^{l}\sum_{i=1}^{N} p_i d_{il} \leq t_l \quad ; l = 1, \dots, L \tag{14}$$

In the case where the processing times P_i are constant and independent of the processed quantity, this condition becomes:

$$\sum_{i=1}^{N}(P_i\left\lceil\frac{\sum_{j=1}^{l} d_{il}}{Q_i}\right\rceil \leq t_l \quad ; l = 1, \dots, L \tag{15}$$

Notice that (15) is a necessary and sufficient condition for the feasibility of Model 2.

4- A DECOMPOSITION SOLUTION APPROACH

A first approach to solve the above introduced problem consists of decomposing the problem into two sub-problems to be solved consecutively. The first sub-problem is the one of determining the product lots to be processed before each delivery date without determining their sequence of production (i.e., without specifying their position *p*). The second sub-problem is to determine the sequence for the obtained lots. To formulate the first sub-problem we can use the following decision variables:

x_{il}	a non-negative integer that indicates the number of lots of product *i* to finish after t_{l-1} and before or up to t_l
q_{il}	the total quantity of product *i* to process and finish after t_{l-1} and up to t_l
I_{il}	inventory level of product *i* at t_l just after delivering the demand d_{il}

Recall that Model 1 allows for more than one lot of the same product *i* to be finished between t_{l-1} and t_l but in different positions. That is why the decision variable x_{il} used to formulate this first sub-problem is an integer that may take a value larger than 1 to allow for more than one lot of *i* to be finished between t_{l-1} and t_l. This first sub-problem can be formulated as follows:

***MODEL* 3: Determination of the lots to process**

Find $x_{il}\in\{0,1,2, ..\}$, $q_{il} \geq 0$, and $I_{il} \geq 0$; i=1,..,N; l=1,..,L, which:

$$Minimize: \sum_{i=1}^{N}\sum_{l=1}^{L} h_i I_{il-1}\,(t_l - t_{l-1}) + \sum_{i=1}^{N}\sum_{l=1}^{L} c_i x_{il} \tag{16}$$

$$Subject\ to: \ q_{il} \le x_{il} Q_i \qquad ; i = 1, \dots, N;\ l = 1, \dots, L \tag{17}$$

$$I_{il} = I_{i,l-1} + q_{il} - d_{il} \qquad ; i = 1, \dots, N;\ l = 1, \dots, L-1 \tag{18}$$

$$F_i = I_{i,L-1} + q_{iL} - d_{iL} \qquad ; i = 1, \dots, N \tag{19}$$

$$\sum_{j=1}^{l}\sum_{i=1}^{N} s_i x_{ij} + p_i q_{ij} \le t_l \qquad ; l = 1, \dots, L \tag{20}$$

The first term in the objective function (16) gives the inventory holding cost of products over the time intervals t_{l-1} and t_l. The second term gives the setup cost of the processed lots. Constraints (17) make sure that the produced quantities do not exceed the upper limit on lot sizes. Constraints (18) and (19) determine the inventory levels and assure that demands are fulfilled without backlogs. Finally, constraints (20) make sure that we have enough time to produce the required quantities (capacity constraints).

This model is linear and contains NL binary variables, $2NL$ continuous variables and $2NL+L$ constraints. Thus if we have 10 products and 20 delivery dates our model has 200 binary variables, 400 continuous variables and 420 linear constraints. It does not contain any non linear functions or constraints. Obviously it is much easier to solve Model 3 than solving Model 1.

In the case where lot processing times (including set up time), denoted P_i, are independent of the quantity to be produced, constraints (20) should be replaced by the following constraint:

$$\sum_{j=1}^{l}\sum_{i=1}^{N} P_i x_{ij} \le t_l \qquad ; l = 1, \dots, L \tag{21}$$

Once the optimal solution of this first sub-problem is obtained we need to solve a second sub-problem to determine the starting date for each of the lots resulting from this solution. This second sub-problem is formulated as the problem of sequencing a set of M jobs (lots) where $M = \sum_{i,l} x_{il}$ and each job consists of processing a number of units (or a lot) of a given product while respecting a given due date.

The optimal solution of the first sub-problem indicates for each product i the number of lots x_{il} to finish by each due date t_l. If x_{il} =1, the corresponding job is to produce q_{il} units of product i and its due date is t_l. In the case where x_{il} >1, we need to determine the size of each of the corresponding x_{il} jobs. All these jobs will be of size Q_i except the first one which will have the size q_{il}-(x_{il}-1)Q_i and all these x_{il} jobs have the same due date t_l.

To simplify the presentation of the second sub-problem, let us renumber the entire set of jobs from 1 to M and use: q_k to denote the size (the number of units to process) of job k (k=1, .., M), t_k to denote its due date, P_k to denote its processing time (including setup time), f_k to denote its finish time and h_k to denote the unit inventory holding cost of the corresponding product. Also, let us use a binary variable, y_{kp} which takes the value 1 if job number k is in position p in the production sequence.

The second sub-problem can then be formulated as follows:

***MODEL* 4: Sequencing the jobs and determination of finish dates**

Find: $y_{kp} \in \{0,1\}$, and $f_k \geq 0$; k=1,..,M, p=1,..,M which:

$$Minimize: \sum_{k=1}^{M} h_k q_k (t_k - f_k) \tag{22}$$

$$Subject\ to: \sum_{k=1}^{M} y_{kp} = 1\,; p = 1, .., M \tag{23}$$

$$\sum_{p=1}^{M} y_{kp} = 1\,; k = 1, .., M \tag{24}$$

$$f_k - f_j - P_k \geq H\left(y_{j,p-1} + y_{kp} - 2\right); j = 1, .., M; k \neq j; p = 2, .., M \tag{25}$$

$$f_k \leq t_k; k = 1, .., M \tag{26}$$

The objective function (22) seeks to minimize the inventory holding cost of the produced items between their finishing time and due dates. Constraints (23) assure that there is only one job in each position of the sequence while constraints (24) are to assure that there is only one position for each job. Constraints (25) make sure that, for each job, the starting time is larger than or equal to the finish time of its immediate predecessor. Constraints (26) restricts the finish time to be less than or equal to the corresponding due time.

The total cost of the production plan produced by this decomposition approach is the sum of the optimum values of models 3 and 4.

Unfortunately, the undertaken numerical tests revealed that it is very time consuming to solve this model 4 as the number of binary variables is relatively large and it has a weak LP relaxation. Consequently, it was decided to solve a restricted version where the position of a job k in the sequence is limited to be between two bounds. The lower bound equals the number of jobs with a due date less than its due date and the upper bound equals the lower bound plus the number of jobs having the same due date. But even with this restriction the model still requires very large computation time. Thus it was decided to limit the computation time to one hour.

Let us notice that the solution obtained by this proposed decomposition approach will often be different from the optimal solution of Model 1. Thus this decomposition approach is actually a heuristic approach.

Numerical illustration

Let us consider the 3-product, 4-delivery date example presented in Table 1 in which lot sizes are bounded and processing times do not depend on the number of units to process. The optimal solution of Model 3 for this numerical example is to produce 2 lots of product 1 and 2 lots of product 2 before time t_1=40(i.e., x_{11}=2 and x_{21}=2), 1 lot of product 3 before t_2=50 (i.e., x_{32}=1), 1 lot of product 1 and 1 lot of product 2 before t_3=70 (i.e., x_{13}=1 and x_{23}=1), and 1 lot of product 3 before t_4=80 (i.e., x_{34}=1). The size of all these lots is the largest possible size except for the two lots of product 1 that should be produced before t_1=40. The total quantity of product 1 to produce q_{11}=100. So we have to produce two lots of a total of 100 units. It is obvious that to minimize the inventory holding cost we should produce these units as late as possible which imply that the second lot should be of the largest possible quantity (80 units as Q_1=80) and the size of the first lot is the difference between the total quantity q_{11} and the size of the second lot (thus of size 20). The minimum value of Model 3 objective function for this example is 3600.

Finally the production sequence obtained by solving of Model 4 for the required 8 jobs (lots) is: 1-2-3-4-5-6-7-8. The corresponding optimum value of the objective function (22) is 904. Table 2 presents the solution obtained and Figure 1 gives the evolution of the inventory level of the 3 products. The total cost as obtained by this decomposition approach is (3600+904=) 4504.

Table 1: numerical example

product	Quantity to be delivered at				Maximum lot size	Processing time/lot	Fixed processing cost	Unit inventory holding cost
	t_1=40	t_2=50	t_3=70	t_4=80				
1	80	0	100	0	80	12	150	0.3
2	0	80	100	120	100	8	250	0.2
3	0	40	0	60	50	10	400	1.0

Table 2: solution of the numerical example

Lot number	1	2	3	4	5	6	7	8
Product number	1	1	2	2	3	1	2	3
Quantity to produce	20	80	100	100	50	80	100	50
Starting time	0	12	24	32	40	50	62	70
Finish time	12	24	32	40	50	62	70	80

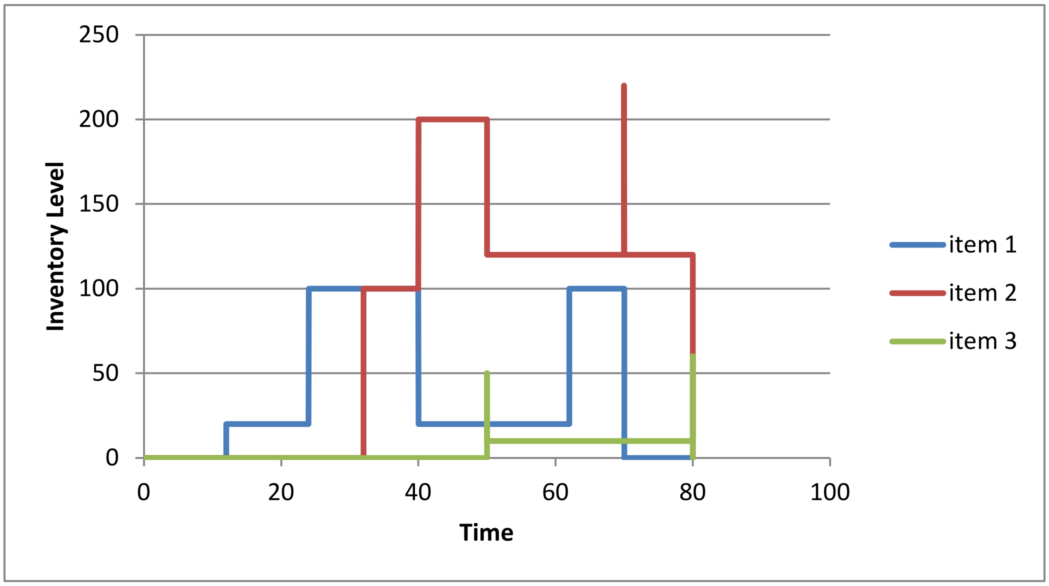

Figure 1: Inventory level for the numerical example

Finally, as that this decomposition approach requires very large computation time, another heuristic which is faster and easier to implement is presented in the next section.

5- A HYBRID HEURISTIC

The proposed heuristic is a hybrid one composed of a solution construction procedure followed by 3 improvement procedures. Figure 2 presents a pseudo-code of this heuristic. The construction phase is an iterative, backward-pass heuristic. To construct the production plan the heuristic starts by setting $t = t_L$, the latest delivery date. The main iteration of the heuristic is as follows. At each date t we list the products to deliver at this date and chose from this list the product k that has the largest value of $h_k q_k$ where q_k= min $\{d_{kt},Q_k\}$. Then we schedule the processing of a lot of size q_k of the product k to finish at t. If q_k=d_{kt} we remove product k from our list otherwise we reduce its demand by q_k. Next, we set $t = t$-P_k, i.e., put t equal the starting time of the just scheduled job k, and add to the list all the orders for which the due date is between t and $t+P_k$ if any. If the resulting list contains more than one order for a given product, we group them into one order. Now, if the list is not empty, repeat the above and choose a job to schedule. Otherwise, move backward to the latest date where we have some orders to deliver. The heuristic stops when there are no more orders to schedule.

The rational of this construction heuristic is to schedule the production of the orders to deliver at the latest possible time in order to minimize the sum of inventory holding costs.

This procedure may produce a non-feasible schedule where the starting time of some jobs is before the beginning of the planning horizon (date 0). Even in such a case we apply the improvement procedures as they may modify this solution in a way that makes it feasible.

The first improvement procedure attempts to reduce the number of production lots in order to reduce the number of setups. The procedure considers one product at a time and repeat the following until no more gain can be achieved. For each lot of the considered product, determine if a gain can be made by removing this lot and adding its units to the preceding lots of the same product. The lot leading to the highest gain is removed and we repeat the same for the remaining lots. This improvement procedure stops if we cannot achieve any more gain.

The second improvement procedure attempts to move the remaining production lots to the latest possible date without causing any backlogs. This may be possible as the previous improvement procedure may remove some production lots making room for processing other lots in the time interval originally occupied by some of the removed lots.

The third improvement procedure exchanges the position of pairs of production lots as long as this can lead to cost reduction without backlogs.

Applied to the numerical example presented in Table 1, the hybrid heuristic produced exactly the same solution as the decomposition approach with a total cost of 4504.

Backward pass construction heuristic
Set $t = t_L$
WHILE *some orders are not scheduled*
List the set of products to deliver at t
Schedule the product k having the largest value of $h_k q_k$ where: $q_k=\min\{Q_k,d_{kt}\}$
IF $d_{kt}=q_k$ THEN remove product k from the list ELSE set: $d_{kt}=d_{kt}-q_k$
Set $t= t-P_k$ where P_k is the processing time of job k
Add to the list all the products to deliver between t and $t+P_k$, if any
IF the list is empty THEN Set: $t=max\{t_l\ |t_l \le t\}$
END WHILE

First improvement procedure: *lot grouping*
REPEAT until no improvement is achieved for any product
FOR each product DO:
REPEAT until no more cost reduction can be achieved
FOR each lot of the considered product :
Determine if a cost reduction can be obtained by removing the considered lot and adding its units to the preceding lots
IF a cost reduction can be achieved AND the gain is larger than the best gain obtained so far:
Store the value of this gain as the best found so far
Store the corresponding modification of the production plan
END IF
NEXT
Remove the lot that allow the best cost reduction, if any, and modify the production plan accordingly
END REPEAT
NEXT
END REPEAT

Second improvement procedure: *postponing lots without causing any backlogs*
FOR each job (lot to be produced) DO:
Postpone the considered job to the latest possible time which does not lead to a backlog
NEXT

Third improvement procedure: *position exchange*
REPEAT until no more cost reduction can be achieved :
FOR each pair of jobs DO:
Permute their positions in the sequence if this does not cause any backlogs and leads to a cost reduction
NEXT
END REPEAT

Figure 2: Pseudo-code of the proposed hybrid heuristic

6- NUMERICAL EXPERIMENT

To assess its performance, the solutions of the hybrid heuristic were compared to those obtained by the decomposition approach using 100 randomly generated problem instances. Unfortunately, it is not possible to obtain the optimal solutions of these instances. In addition the literature does not provide any solution method that can be used to assess the performance of the proposed heuristics.

For all the generated instances, the number of products is 10 and the number of delivery dates is 8. Delivery dates are: 40, 60, 80, 100, 120, 150, 180 and 200. For each instance, the values of h_i, c_i, P_i and Q_i are randomly drawn for uniform distributions. The limits for these uniform distributions are respectively from 0.05 to 0.25 for h_i, from 80 to 200 for c_i, from 1 to 5 for P_i, and from 40 to 80 for Q_i. To determine the total demand of a product, a random value is drawn from a uniform distribution between 120 and 180. Then the number of its delivery dates is randomly drawn from a uniform distribution between 1 and 4. The total demand is then partitioned and a quantity is randomly determined for each delivery date. Also, the necessary and sufficient feasibility condition (15) is used to make sure that each instance has a feasible solution.

Solutions obtained by the decomposition approach

The decomposition procedure consists of solving Model 3 to determine the lots to produce for each product and each due date. Then we have to solve Model 4 to determine the production sequence of these lots. The total cost is determined by adding the costs given by Models 3 and 4. A computer equipped with 2.66 GHz, 2 cores Intel Xeon processor and the commercial MIP code GORUBI 5.5.0 was used to solve Model 3. This required an average of 1.261 CPU seconds.

Unfortunately, solving Model 4 is revealed to require very long computational times. Actually, after several days of computation time without obtaining the optimal solution for the first instance, it was decided to put one hour limit on the computational time. No instance was solved to optimality within this run time limit and no feasible solution was found for 71 of the 100 instances. The total cost of the for the remaining 29 problem instances (for which a feasible solution for Model 4 was obtained) is 18.9% higher than the cost of the solution for these instances given by the hybrid heuristic.

Then it was decided to increase the run time limit to 3 hours. Again, no instance was solved to optimality within this run time limit and no feasible solution was found for 70 of the 100 instances. The total cost of the for the remaining 30 problem instances (for which a feasible solution for Model 4 was obtained) is 16.80% higher than the cost of the solution for these instances as obtained by the hybrid heuristic.

Finally, it was decided to solve a restricted version of Model 4 hoping that a relatively good solution could be found in a reasonable computation time. In this restricted version, the position of a job k in the sequence is restricted to be between two bounds. The lower bound equals the number of jobs with a due date less than its due date and the upper bound equals the lower bound plus the number of jobs having the same due date. In other words, no job can be positioned before any of those with earlier due date.

Unfortunately, after several hours of computation time, we failed to solve the first problem to optimality. Consequently, it was decide to limit the computation time to one hour. Only two instances were solved to optimality within this time limit. Nevertheless, a feasible integer solution was found for all the other 98 instances.

Solutions obtained by the simple hybrid heuristic

The proposed hybrid heuristic succeeded to solve all test instances with an average time of less than 1 second. The 3 improvement procedures reduce the total cost obtained at the construction step by 8.01% in average with a minimum improvement of 2.12% and a maximum improvement of 16.76%. In more details, the first improvement procedure, the grouping procedure, produced an improvement of 4.42% in average while the second and third improvement procedures yielded 1.05% and 2.73% improvement in average.

The hybrid heuristic produced a better solution than the decomposition approach for 75 instances while the decomposition approach produced a better solution for the other 25 instances. Over the entire 100 instances set, in average, the hybrid heuristic produced solutions with a total cost 3.38% less than the decomposition approach.

7- CONCLUSIONS

To the best of my knowledge, this paper is the first one to deal with the generalized capacitated single-facility lot sizing and scheduling problem (GSCLSP) as defined above. In this problem, we have to produce several products sharing the same production facility, each product has a set of delivery dates and the quantity to deliver at each of these dates is known. Setups can be carried over if the same product is processed in consecutive lots. Also the inventory holding cost of the units produced within a given lot is calculated from its finish date up to the delivery of all its units.

Two heuristic approaches are proposed: the decomposition approach and a hybrid heuristic. The decomposition approach consists of solving a first mathematical model to determine the lots to process and a due date for each one. The second model is designed to produce a feasible schedule for the processing of these lots. While it was revealed easy to solve the first model, the second one cannot be solved in a reasonable amount of computational time. Thus it was necessary to reduce its solution space and to put a limit on its solution time in order to get feasible solutions. However this restriction may lead to solutions of higher costs than if we solve the non-restricted model to optimality.

Although, the hybrid heuristic is very fast and produces relatively good solutions, it is clear that we need to develop even better methods to solve the GSCLSP. We also need to develop solution methods for the multiple-facility version of the problem as well as the multistage one.

Acknowledgement

This research work was partially supported by grant OPG0036509 from the National Science and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.

REFERENCES

Boctor, F.F. (1987), The G-group heuristic for single-machine lot scheduling. *International Journal of Production Research*, 25, 363-379.

Brahimi, N., Dauzere-Peres, S., Najid, N.M. and Nordli, A.(2006), Single item lot sizing problems. *European Journal of Operational Research*, 168, 1–16.

Delporte, C., and Thomas, L. (1978), Lot sizing and sequencing for N products on one facility. Management Science, 23, 1070–1079.

Dobson, G. (1987), The economic lot-scheduling problem: Achieving feasibility using time-varying lot sizes. Operations Research, 35, 764–771.

Jans R. and Degraeve Z. (2008), Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46, 1619-1643.

Gicquel C., Minoux M., and Dallery Y. (2008), Capacitated lot sizing models: a literature review (http://hal.archives-ouvertes.fr/docs/00/25/58/30/PDF/CER_Gicquel_2008_01.pdf) consulted November 2012.

Buschkühl L, Sahling F, Helber S and Tempelmeier H. (2010), Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. *OR Spectrum*, 32, 231-261.

Elmaghraby, S.E. (1978), The economic lot scheduling problem (ELSP): Review and extensions. *Management Science*, 24, 587-598.

Gopalakrishnan, M., Miller, D.M. and Schmidt, C.P. (1995), A framework for modeling setup carryover in the capacitated lot sizing problem. *International Journal of Production Research*, 33, 1973-1988.

Gopalakrishnan, M. (2000), A modified framework for modeling set-up carryover in the capacitated lot sizing problem. *International Journal of Production Research*, 38, 3421–3424.

Gopalakrishnan, M., Ding, K., Bourjolly, J.-M. and Mohan, S. (2001), A Tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Management Science*, 47, 851–863.

Porkka, P., Vepsalainen, A.P.J. and Kuula, M. (2003), Multi-period production planning carrying over set-up time. *International Journal of Production Research*, 41, 1133–1148.

Raza, A. R. and Akgunduz A. (2008), A comparative study of heuristic algorithms on the economic lot scheduling problem, *Computers and Industrial Engineering*, 55, 94-109.

Sox, C.R. and Gao, Y. (1999), The capacitated lot sizing problem with setup carry-over. *IIE Transactions*, 31, 173–181.