International Workshop on Lot Sizing (IWLS) 2015
Montréal, Canada

HEC Montréal

August 24-26, 2015

Local Organizing Committee:

Raf Jans, HEC Montréal, Canada (raf.jans@hec.ca)
Jean-François Cordeau, HEC Montréal, Canada (jean-francois.cordeau@hec.ca)
Ola Jabali, HEC Montréal, Canada (ola.jabali@hec.ca)
Yossiri Adulyasak, HEC Montréal, Canada (yossiri.adulyasak@hec.ca)

Steering Committee:

Nabil Absi (France)
Bernardo Almada-Lobo (Portugal)
Christian Almeder (Germany)
Stéphane Dauzère-Pérès (France)
Silvio de Araujo (Brazil)
Meltem Denizel (Turkey)
Stefan Helber (Germany)
Hark-Chin Hwang (South Korea)
Safia Kedad-Sidhoum (France)
Edwin Romeijn (USA)
Wilco van den Heuvel (Netherlands)
Laurence A. Wolsey (Belgium)

Supported by:

Group for Research in Decision Analysis (GERAD) (www.gerad.ca)
HEC Montréal (www.hec.ca)
Association of European Operational Research Societies (EURO) (www.euro-online.org)
Canada Research Chair in Logistics and Transportation
HEC Montréal Professorship in Operations Planning

Printed in Montréal — August 2015

List of Participants

Name		**Affiliation**	**Country**
Nabil	Absi	Ecole des Mines de Saint-Etienne	France
Andrés	Acevedo	Concordia University	Canada
Yossiri	Adulyasak	HEC Montréal	Canada
Kerem	Akartunali	University of Strathclyde	UK
Bernardo	Almada-Lobo	INESC-TEC, Porto University	Portugal
Christian	Almeder	European University Viadrina Frankfurt (Oder)	Germany
Pedro	Amorim	INESC TEC	Portugal
Tiffany	Bayley	University of Waterloo	Canada
Hatem	Ben Amor	AD OPT	Canada
Yuan	Bian	Ecole des Mines de Nantes	France
Jim	Bookbinder	University of Waterloo	Canada
Canan	Capa	Concordia University	Canada
Masoud	Chitsaz	HEC Montréal	Canada
Ivan	Contreras	Concordia University	Canada
Karina	Copil	University of Cologne	Germany
Jean-François	Cordeau	HEC Montréal	Canada
Stéphane	Dauzère-Pérès	Ecole des Mines de Saint-Etienne	France
Sascha	Düerkop	University of Cologne	Germany
Ioannis	Fragkos	HEC Montréal	Canada
Grigori	German	Univ. Grenoble Alpes	France
Matthieu	Gruson	CIRRELT	Canada
Fanny	Hein	European University Viadrina Frankfurt (Oder)	Germany
Stefan	Helber	Leibniz Universität Hannover	Germany
Timo	Hilger	University of Cologne	Germany
Ola	Jabali	HEC Montréal	Canada
Raf	Jans	HEC Montréal	Canada
Safia	Kedad-Sidhoum	LIP6 - UPMC	France
Onur Alper	Kilic	Hacettepe University	Turkey
Alf	Kimms	University of Duisburg-Essen	Germany
Svenja	Lagershausen	Leibniz University Hannover	Germany
Andrea	Lodi	École Polytechnique de Montréal	Canada
Xue	Lu	London Business School	United Kingdom
Gislaine Mara	Melega	CIRRELT	Canada
Srimathy	Mohan	Arizona State University	USA
Siao-Leu	Phouratsamay	Lip6, UPMC	France
Eric	Prescott-Gagnon	JDA Software	Canada
Vincent	Raymond	JDA Software	Canada
Steffen	Rickers	Leibniz Universität Hannover	Germany
Florian	Sahling	Leibniz Universität Hannover	Germany
Katja	Schimmelpfeng	University of Hohenheim	Germany
Narges	Sereshti	Dalhousie University	Canada
Xueying	Shen	DecisionBrain	France
Dariush	Tavaghof Gigloo	TU Munich	Germany
Horst	Tempelmeier	University of Cologne	Germany
Reha	Uzsoy	North Carolina State University	USA
Yasser	Valcárcel Miró	HEC Montréal	Canada
Wilco	Van den Heuvel	Erasmus University Rotterdam	The Netherlands
Albert	Wagelmans	Erasmus University Rotterdam	The Netherlands
Martin	Wörbelauer	Universität Hohenheim	Germany

Bienvenue à Montréal!

Dear colleagues,

It is a pleasure to welcome you to the sixth International Workshop on Lot Sizing, which is held at HEC Montréal. We continue the tradition of the previous workshops to discuss high quality research in a relaxed atmosphere. As in the previous editions, the aim of the workshop is the following:

"To cover recent advances in lot sizing: new approaches for classical problems, new relevant problems, integration of lot sizing with other problems, presentation of case studies, etc. The workshop will also aim at favoring exchanges between researchers and enhancing fruitful collaboration."

We would like to thank our sponsors for their support in organizing this workshop: HEC Montréal, GERAD, EURO, the Canada Research Chair in Logistics and Transportation and the HEC Montréal Professorship in Operations Planning. Special thanks go to the team at GERAD (Marie, Carole and Marilyne) for their valuable support in organizing this workshop.

We wish you a nice stay in Montréal and hope that you find the workshop inspiring and productive.

Raf Jans, Jean-François Cordeau, Ola Jabali and Yossiri Adulyasak

IWLS 2015 Program

Day 1: Monday, August 24, 2015

08:30-09:15 **Registration**

09:15-09:30 **Welcome Session**

09:30-10:30 **Session - Decomposition Approaches (chair: Raf Jans)**

10:30-11:00 **Coffee Break**

11:00-12:30 **Session - Stochastic Lot-Sizing I (chair: Pedro Amorim)**

12:30-14:00 **Lunch**

14:00-15:30 **Session - Heuristics I (chair: Stéphane Dauzère-Pérès)**

15:30-16:00 **Coffee Break**

Day 2: Tuesday, August 25, 2015

A Generalized Dantzig-Wolfe Decomposition Algorithm for Mixed Integer Programming Problems

Xue Lu
London Business School
xlu@london.edu

Zeger Degraeve
Melbourne Business School
Z.Degraeve@mbs.edu

Abstract

We propose a generalized Dantzig-Wolfe decomposition algorithm for mixed integer programming problems. By generating copy variables, we can reformulate the original problem to have a diagonal structure which is amendable to the Dantzig-Wolfe decomposition. We apply the proposed algorithm to the multi-level capacity constrained lot sizing problem and production routing problem. Computational results show that, to the best of our knowledge, our algorithm provides a tighter bound of the optimal solution than all the existing methods.

1 Introduction

A mixed integer programming (MIP) problem is a mathematical optimization problem in which some but not all of the variables are restricted to be integers. The objective is a linear function to be minimized or maximized, and the constraints other than the integer restrictions are linear equalities or/and inequalities. For simplicity, we assume that the objective function is in minimization form in the reminder of this paper. The discrete nature of the integer variables greatly expands the application of MIP. However, the existence of integer variables increases the complexity of a MIP problem exponentially. Solving MIP problems to optimality using exact algorithms is not guaranteed. One stream of research focuses on developing efficient heuristics to generate feasible solutions, which serve as upper bounds of the optimal solution. Heuristics can be problem specific, and we cannot evaluate the quality of any feasible solution without knowing the optimal solution. In this paper, we focus on developing tight lower bounds. A tight lower bound can be used to evaluate the quality of the existing heuristic solutions. More importantly, exact algorithms for MIP problems are lower bound based. Thus a tight lower bound may significantly speed up the convergence of exact algorithms. When the complicated integrality constraints are dropped, a MIP problem turns into a linear programming (LP) problem which can be solved efficiently even with hundreds of thousands of constraints and variables. The optimal solution of the resulting LP problem serves as a lower bound of the optimal solution of the original MIP problem. Although some strong formulations have been proposed in the literature, lower bounds from linear relaxation are generally very far away from the optimal solution. We will propose a generalized Dantzig-Wolfe decomposition algorithm as an alternative approach to compute lower bounds in Section 2, and demonstrate how to apply the proposed algorithm to the multi-level capacity constrained lot sizing problem in Section 3 and production routing problem in Section 4.

2 A Generalized Dantzig-Wolfe Decomposition

Dantzig-Wolfe (DW) decomposition is an algorithm developed to deal with constraint matrix with a special block diagonal structure, where each block is associated with a subset of variables. The idea of DW decomposition is to reformulate the original variables as a convex combination of the extreme points of substructures, by keeping the coupling constraints in the master. When the integrality constraints are dropped in the master, we will achieve a lower bound of the optimal solution. However, the key challenge of applying DW decomposition is that the constraint matrix of a MIP problem does not always have the block diagonal structure. In the following part, we will demonstrate a generalized DW decomposition algorithm which even works on MIP problems with non-block-diagonal constraint matrix.

First we decompose the set of linear constraints into a set of subsystems. Note that the subsystems after decomposition need not be mutually exclusive, i.e., two subsystems may share some constraints.If the constraint matrix has a block diagonal structure, each subset of constraints represents one block; otherwise, there exist variables that appear in multiple subsets of constraints. Next we create a duplicate of such variables for each subset of constraints in which they appear, and add equality constraints to the master to ensure the new generated duplicates have the same value as their original one. For example, if variable x_1 appears in both subset 1 and 2, we will generate a duplicate x_1^d, replace x_1 with x_1^d in subset 2 of constraints, and add $x_1 = x_1^d$ to the master. By doing so, each subset of constraints shares no variables with other subsets, and thus a block diagonal constraint matrix is created. We can prove analytically that the lower bound generated by the proposed generalized DW decomposition is at least as tight as and generally much tighter than that from linear relaxation. Moreover, we can therefore construct a generalizable formal solution procedure, branch-and-price, to solve MIP problems to optimality.

3 Multi-level Capacity Constrained Lot Sizing Problem

We consider the multi-level capacity constrained lot sizing problem (MLCLSP), where the objective is to minimize the total cost, subject to satisfying the demands of multiple products over multiple periods with limited resources. In the literature, DW decomposition has been applied to the single-level capacity constrained lot sizing problem (CLSP): the capacity constraints are kept in the master and each subproblem represent an uncapacitated single item lot sizing problems. After adding the constraints to capture the multi-level product structure, MLCLSP cannot be tackled by the classic DW decomposition any longer. In the following part, we will illustrate two ways to implement the generalized DW decomposition on MLCLSP.

Horizon Decomposition (HD): the whole time period is decomposed into a set of contiguous horizons, with the chosen horizons length and overlapped periods. For example, HD 2-0 represents a HD with horizon length = 2 and no overlaps. The subproblems have the same structure as the original MLCLSP but with shorter time horizons. Equality constraints are added to the master to ensure that any variable in the overlapped periods between two subproblems are equal to each other. We test HD on instances with 10 products over 24 periods. As shown in Table 1, HD with longer horizon and overlapped periods tends to generate tighter lower bound. An explanation is that subproblems with longer horizons and overlapped periods may capture more information of the original problem since more integrality constraints are preserved in the subproblems. The best performing HD 6-1 generates a lower bound that is 50% tighter on average than that from CPLEX within the same CPU time, and 70% tighter than that from linear relaxation of the strong formulation. Note that a tight lower bound is rather expensive in term of CPU time.

Production Decomposition (PD): the multi-level product structure is decomposed into a set of sub structures. The resulting subproblems have the same structure as the original MLCLSP but with fewer products. We test PD on instances with 40 products over 8 periods. Equality constraints are added to the master to ensure that any variable of the overlapped products between two subproblems are equal to each other. As shown in Table 2, the best PD 1-3 achieves a much tighter lower bound within less CPU time than CPLEX. In addition, PD with more products in each subproblem does not necessarily generate tighter lower bound. 1-3, 1-1-3 and 1-1-1-3 are three different ways to decompose the product structure, that 1-3 has 4 products in each subproblem while 1-1-3 has 5. Note that PD 1-1-3 requires almost 10 times as much on the average CPU time as PD 1-3 but fails to generate a tighter lower bound. An explanation is that overlaps on product structure between subproblems might lead to degeneracy during the computing process.

		Intergrality Gap		CPU Time	
		No Overlap	1 Period Overlap	No Overlap	1 Period Overlap
	4	9.26%	7.22%	59s	479s
Horizon	5	8.05%	6.42%	177s	1204s
	6	6.92%	6.37%	429s	1210s

Table 1: MLCLSP with Horizon Decomposition (CPLEX Gap = 13%, 1200s)

	Integrality Gap			CPU Time		
	1-3	1-1-3	1-1-1-3	1-3	1-1-3	1-1-1-3
Mean	8.09%	8.32%	8.51%	220s	2095s	299s

Table 2: MLCLSP with Product Decomposition (CPLEX Gap = 12%, 1200s)

Integrality Gap	No. of Customers = 4,6,8,10					
	Linear Relaxation			Dantzig-Wolfe Decomposition		
No. of Periods = 2,...,100	Miller-Tucker-Zemlin	Subtour Elimination	One-Commodity Flow	B&N (2010)	HD 1-0	HD 2-0
Mean	48.29%	26.95%	20.86%	16.91%	9.68%	7.72%

Table 3: Production Routing Problem with Horizon Decomposition

4 Production Routing Problem

The production Routing Problem (PRP) combines the lot sizing problem and vehicle routing problem, and jointly optimize the supply chain over production, distribution and inventory. [1] gives a detailed review on PRP discussing various formulations and solution algorithms. Bard and Nananukul (B&N) proposed a DW decomposition in which the production planning and vehicle routing are separated [2]. We apply HD to PRP, and test instances with 4 to 10 customers over 2 to 100 periods. Computational results are summarized in Table 3. On average, HD 2-0 generates a lower bound with 7.72% integrality gap, 20% tighter than that from HD 1-0. This confirms our finding in Section 3 that HD with longer horizon tends to generate tighter lower bound. On average, the lower bound generated by HD 2-0 is 54% tighter than the lower bound from the decomposition approach proposed by B&N [2]. Moreover, we provide the benchmark lower bounds generated by linear relaxing different formulations.

References

[1] Adulyasak, Y., Cordeau, J.F., Jans, R., The production routing problem: A review of formulations and solution algorithms, Computers & Operations Research, 55, 141-152 (2015)

[2] Bard, J.F., Nananukul, N., A branch-and-price algorithm for an integrated production and inventory routing problem, Computers & Operations Research, 37(12), 2202-2217 (2010)

New Lower Bounds for Single-Item Lot-Sizing

Grigori German, Hadrien Cambazard, Bernard Penz, Jean-Philippe Gayon
Univ. Grenoble Alpes, G-SCOP, F-38000 Grenoble, France
CNRS, G-SCOP, F-38000 Grenoble, France
firstname.name@grenoble-inp.fr

Abstract

This paper presents a generic new way of computing lower bounds for the different costs of the single-item lot-sizing problem with production capacities and bounded inventory. The lower bounds can be computed the same way for the production setup cost and the variable production and storage costs, based on a decomposition of the horizon into sub-problems and the use of the Weighted Interval Scheduling Problem. Experimental analysis show that the bound on the global cost can be higher than the ones given by linear programming solvers.

Keywords: lower bound, single-item lot-sizing.

1 Introduction

These lower bounds are motivated by the work currently in progress on a Constraint Programming global constraint for lot-sizing problems by [3].
The single-item lot-sizing problem aims at planning the production of a single type of item over a finite horizon of T periods in order to satisfy a demand d_t at each period t. The production cost at t is defined by a unitary cost p_t (cost per unit) and a setup cost s_t paid if at least one unit is produced a t. The holding cost h_t is paid for each unit stored at the end of period t. Furthermore the production (respectively the inventory) is bounded by minimal and maximal capacities $\underline{\alpha_t}$ and $\overline{\alpha_t}$ (respectively $\underline{\beta_t}$ and $\overline{\beta_t}$) at each period t. The goal is to determine a production plan that satisfies the demands, respects the capacities and minimizes the global cost. The following variables are used to model the problem:

- $X_t, I_t \in \mathbb{N}$: Quantity produced at t and quantity stored between t and $t+1$.
- $Y_t \in \{0, 1\}$: Equals 1 if at least one unit is produced at t, 0 otherwise.
- $C \in \mathbb{R}_+$: Global cost.
- $Cp, Ch, Cs \in \mathbb{R}_+$: Sum of the variable production and holding costs and of the setup costs.

A mathematical model for that problem (L) can be written as follows:

$$
(L)\quad
\begin{array}{lll}
 & \text{Minimize } C = Cp + Ch + Cs & \\
(L.1) & I_{t-1} + X_t - I_t = d_t & \forall\, t = 1 \dots T \\
(L.2) & X_t \leq \overline{\alpha_t} Y_t & \forall\, t = 1 \dots T \\
(L.3) & Cp = \sum_{t=1}^{T} p_t X_t & \\
(L.4) & Ch = \sum_{t=1}^{T} h_t I_t & \\
(L.5) & Cs = \sum_{t=1}^{T} s_t Y_t & \\
(L.6) & X_t \in \{\underline{\alpha_t}, \dots, \overline{\alpha_t}\} I_t \in \{\underline{\beta_t}, \dots, \overline{\beta_t}\}, Y_t \in \{0, 1\} & \forall\, t = 1 \dots T
\end{array}
$$

(L.1) are the flow balance constraints for each period, (L.2) are the setup constraints and (L.3), (L.4) and (L.5) are the expressions of the different costs.

Particular cases of (L) have been broadly studied in the literature depending on the parameters taken into account. In 1958, Wagner and Whitin introduced the original uncapacitated problem in [5] and solved it using dynamic programming in $O(T^2)$. The single-item lot-sizing problems with time-varying production capacities and setup costs are however NP-hard [1].

2 The new lower bound

The general idea of the bound is to decompose (L) into sub-problems then to compute a lower bound on each of them and finally combine them 'at best' to find a global lower bound.

2.1 Production sub-problem

$\forall u, v \in \{1, \dots, T\}$, $u < v$, we define a sub-problem L_{uv} as (L) with:

- $\forall t \notin \{u, u+1, \dots, v\}$, $d_t = 0$
- $\forall t < u$, $s_t = 0$

Hence L_{uv} is equivalent to determine a lower bound of the cost needed to satisfy all the demands in $D_{uv} = \{d_u, \dots, d_v\}$. The set of solutions satisfying these demands is dominated by the solutions where $I_v = 0$ – every solution has a lower cost if nothing is stored at the end of period v. Moreover, some demands in D_{uv} can be satisfied by a production previous to the period u. Yet computing the optimal cost of obtaining a given storage level (in the interval $\{0, \dots, \min(\sum_{t=u}^{v} d_t, \overline{\beta_{u-1}})\}$) for I_{u-1} can be achieved with an easy greedy algorithm: we just have to determine the cheapest periods – without the setup cost – in order to produce the given quantity of units and store it until u. Finding an optimal production plan – or a lower bound on its cost – of L_{uv} gives a lower bound on the cost needed to satisfy all the demands in D_{uv} in (L).

We define the size of a sub-problem by $s_{uv} = (v - u + 1) I_{max}^{uv}$ where $I_{max}^{uv} = \max\{\overline{\beta_t} | t \in \{u, \dots, v\}\}$.

2.2 Weighted Interval Scheduling Problem: a generic way to combine sub-problems

We propose here a dynamic programming approach to compute bounds on the cost variables. Let C_v be a cost variable of (L) (C_p, C_h, C_s, or C). The idea is to associate a lower bound of that cost to each sub-problem and find a set of disjoint sub-problems that maximizes the sum of these values.

All the sub-problems (there are $\frac{T(T-1)}{2}$ of them) are ordered by increasing end times first, then by increasing start times: $[1,2],[1,3],[2,3],[1,4],[2,4],[3,4],\ldots,[T-1,T]$. For the i^{th} sub-problem corresponding to the interval $[u,v]$, we denote w_i the lower bound of its cost, i.e. $w_i = w_{u,v}$. Let's note that any set of disjoint sub-problems gives a lower bound of C_v by considering the sum of the w_i associated to these disjoint sub-problems.

By this principle, in order to determine the best lower bound on C_v, we want to determine a **set S of disjoint sub-problems that maximizes $\sum_{k\in S} w_k$**. That is precisely the Weighted Interval Scheduling Problem [4], solved polynomially in the general case in $O(n\log(n))$ (n being the number of intervals), using a sort on the intervals in $O(n\log(n))$ and a dynamic programming algorithm that runs in $O(n)$. In our case, the intervals are already sorted and the dynamic programming algorithm runs in $O(T^2)$. We consider the given order of the intervals and denote $f^*(i)$ the maximal weight that can be achieved using the i first intervals. With $f^*(0)=0$, we have $\forall i = 1,\ldots,\frac{T(T-1)}{2}$:

$$f^*(i) = \max(f^*(i-1), f^*(p_i) + w_i)$$

where p_i is the biggest integer, smaller than i ($p_i < i$), such as the intervals p_i and i are disjoint. Hence p_i is the first interval before the i^{th} one that is time-compatible with it. For instance, $[3,4]$ is the 6^{th} interval and $p_6 = 1$ since $[1,2]$ is time-compatible with $[3,4]$ and $[1,3]$ is not. Therefore the value $C_{dyn} = f^*(\frac{T(T-1)}{2})$ is a lower bound of C_v.

2.3 Examples of lower bounds on C

Any bound that can be computed on a sub-problem can be of use. Especially bounds that are too time-consuming to compute on the whole horizon can be restricted to smaller sub-problems. Here a two examples of bounds that can be used:

LBDyn: The dynamic programming algorithm of [2] runs in $O(TI^{uv}_{max})$ which is not polynomial and thus cannot be used to find a lower bound on each sub-problem. However, solving it on all the sub-problems of reasonable size gives the best lower bound on these sub-problems.

LBFlow: Solving the linear relaxation of (L) – $Y_t \in [0,1], \forall t = 1\ldots T$ - can be shown to be equivalent to a flow problem that can be solved in polynomial time and without the use of a linear programming solver. Solving it on the sub-problems for which the dynamic programming algorithm is too slow gives lower bounds on the global cost of these sub-problems.

3 Experimental analysis

The following experimentation was conducted on the set of instance classes introduced in [3]. Table 1 compares the linear relaxation, and the root node bound of the aggregated linear programming model solved with Cplex to our bound. The latter has been computed four times. The lower bound used for the smaller sub-problems ($s_{uv} < sMax$) is LBDyn and LBFlow was used on the bigger sub-problems. The gap to the optimum has been averaged for each class and is compared here. The second column is the size of L_1T for the class of instances. We can see that if all the sub-problems use only LBFlow (*i.e.* $sMax = 0$), we obtain the same bound as the linear relaxation. The execution time for the linear models are under $1s$. The computation times (in seconds in table 1) are quite high, however it gives an idea of what bounds could be obtained using good efficient bounds on only the smaller sub-problems.

		AGG RNB	AGG LR	$sMax = 0$		$sMax = 2.10^4$		$sMax = 3.10^4$		$sMax = 5.10^4$	
Class	Size	Gap Opt	Gap Opt	Gap Opt	CPU	Gap Opt	CPU	Gap Opt	CPU	Gap Opt	CPU
C1	1.10^5	3.13%	10.67%	10.67%	0.62	7.91%	21.47	4.81%	52.91	1.77%	135.22
C2	1.10^5	1.43%	8.83%	8.83%	0.43	5.46%	21.88	2.93%	51.21	1.23%	119.33
C3	2.10^5	2.32%	9.79%	9.79%	2.60	8.28%	46.13	4.30%	116.95	2.12%	262.06
C4	2.10^5	1.77%	9.03%	9.03%	2.79	6.04%	45.03	3.60%	131.82	1.79%	256.75
C5	2.10^5	1.43%	9.37%	9.37%	4.00	5.99%	47.89	3.46%	141.25	1.59%	263.00

Table 1: Experimental results

4 Conclusion

We presented here a generic way to compute lower bounds on each cost of the single-item lot-sizing problem with production capacities and bounded inventory. The use of this bound in Constraint Programming is double since, not only it provides good lower bounds, the way it is computed might allow an efficient filtering of the variables.

References

[1] G. R. Bitran and H. H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186, 1982.

[2] M. Florian and M. Klein. Deterministic production planning with concave costs and capacity constraints. *Management Science*, 18(1):12–20, 1971.

[3] G. German, H. Cambazard, J.-P. Gayon, and B. Penz. Une contrainte globale pour le lot sizing. *Accepted at JFPC 2015 in Bordeaux, France*, 2015.

[4] J. Kleinberg and É. Tardos. *Algorithm design.* Pearson Education India, 2006.

[5] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. *Management science*, 5(1):89–96, 1958.

Stochastic Capacitated Lot-Sizing under Consideration of Customer Order Waiting Times

Timo Hilger and Horst Tempelmeier
University of Cologne

Abstract

We consider a stochastic dynamic capacitated lot-sizing problem with a constraint on the customer order waiting times. We propose a MIP model that allows to study the effect of the lot-sizes on the probability distribution of the waiting times. A numerical study outlines that the costs and inventory levels as well as the capacity requirements are significantly driven by the customer order waiting time constraints. Additionally, it is shown that the waiting time distribution in a supply chain may have a significant impact on the performance of downstream nodes in a supply chain.

1 Introduction

A key success factor for managing supply chains efficiently is a controllable lead time, respectively customer order waiting time. The customer order waiting time is defined as the time interval between order arrival and its delivery from the warehouse to the customer (see [5]). The waiting time distribution of a manufacturer significantly affects the performance of downstream nodes within a supply chain, influences the customers' evaluation of the service offered by a company and can be used to manage logistical processes in supply chains more efficiently. Lot-sizing problems with a constrained customer order waiting time is not covered by the existing literature. The present contributions on customer order waiting time focus on inventory management topics. Therefore, a mixed integer approximation model of the Stochastic Multi-Item Capacitated Lot-Sizing Problem (SMICLSP) is proposed that allows to constrain the customer order waiting time and to quantify the monetary consequences.

2 Problem Statement and Modeling Approach

The objective is to generate a production plan that minimizes the sum of the expected setup and inventory holding costs over the planning horizon of T periods ($t \in \{1, 2 \ldots, T\}$) and for K products ($k \in \{1, 2 \ldots, K\}$) by guaranteeing a minimum service level supplied within a predefined customer order waiting time. The expected demands $E\{D_{kt}\}$ as well as the variances $VAR\{D_{kt}\}$ are known as a result of the applied forecasting system. It is assumed that the static uncertainty strategy (see [1]) is applied. At the beginning of the planning horizon the complete production plan is fixed, including the period and the amount of the production quantities.

Figure 1 illustrates the relationship of the duration of a stock-out situation and the waiting time. The production quantity q raise the physical inventory I^p, which is used to fulfill the customer demand. If the physical inventory is not sufficiently high, the customer demands cannot be served and are backordered until the arrival of the next replenishment. Thus, the occurring backorders are characterized by different discrete waiting times. For instance, the backordered demand in period four can be served with the production quantity q in period seven. Therefore, the resulting waiting time is $wt = 3$. Hence, the waiting time for the backordered demand in period five and six is $wt = 2$, respectively $wt = 1$. In case of multiple replenishment cycles, different waiting times wt can be observed repeatedly. The customer order waiting time can be controlled in stochastic lot-sizing models by determining the lot-sizes and the length of the production cycles.

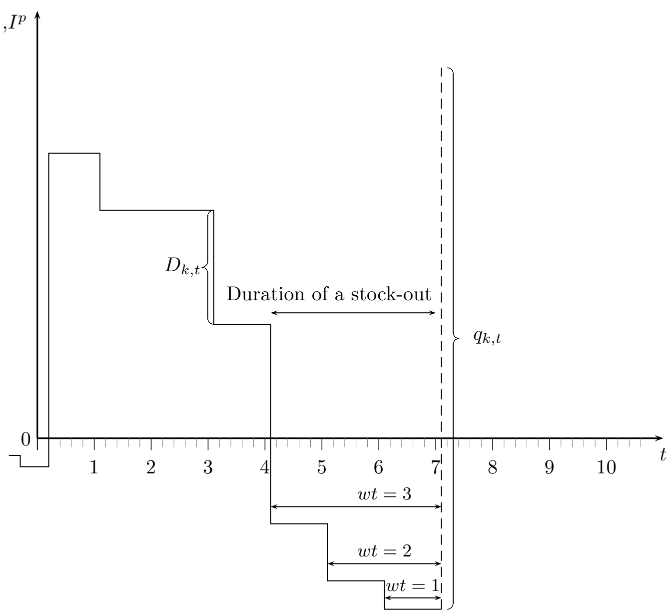

Figure 1: Duration of a stock-out and the customer order waiting time

An approximation of the probability distribution of the customer waiting time for a discrete time model and a given production plan has been proposed by [2]. It is assumed that the management can specify the maximum of the accepted waiting time, $wt^{\max}$, and set minimum service levels for the proportion of the demand that is fulfilled without any delay directly from stock and the service supplied after $wt^{\max}$ periods. The formulation of these conditions specify a lower bound, as only one of the service-constraints may be binding for the optimization. The non-linear model formulation containing the customer order waiting time constraints is solved by a deterministic mixed-integer linear program using a piecewise-linear approximation of the non-linear functions of the expected values (see [3], [6] and [4]).

3 Numerical Results

A simulation study indicate the high approximation quality and shows that 100% of the simulated production plans miss one or both of the predefined service conditions by at most 0.5 percentage points.

Further, a numerical study reveals that the local costs and inventory levels as well as capacity requirements of the manufacturer are significantly driven by the customer order waiting time distribution. The lower the agreed service conditions and the longer the accepted limits of the customer order waiting time are, the higher are the cost saving potentials of the manufacturer. A remarkable result is that the total costs can be reduced significantly for the service supplied within the maximum accepted waiting time if the service agreement for direct demand fulfillment is reduced. Additionally, it is shown that the waiting time distribution of a manufacturer significantly affects the performance of downstream nodes in a two-level supply chain. The longer the observed average waiting time and the greater the randomness of the lead time, the higher are the safety stock requirements of the inventory node. It is shown that the total costs in a centrally controlled two-level supply chain can be reduced. However, the manufacturing costs typically tend to outweigh the inventory holding costs if the inventory holding costs of both echelons are similar.

References

[1] Bookbinder JH, Tan, JY (1988): Strategies for the Probabilistic Lot-Sizing Problem with Service-Level Constraints *Management Science (9)*, 1096–1108

[2] Fischer L (2008): Bestandsoptimierung für das Supply Chain Management – Zeitdiskrete Modelle und praxisrelevante Ansätze *Norderstedt:Books on Demand*

[3] Helber S, Sahling F, Schimmelpfeng (2013): Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR Spectrum (35)*, 75–105

[4] Rossi R, Kilic O, Tarim (2015): Piecewise linear approximations for the static–dynamic uncertainty strategy in stochastic lot-sizing. *Omega (50)*, 126–140

[5] Tempelmeier H, Fischer L (2010): Approximation of the probability distribution of the customer waiting time under an (r, s, q) inventory policy in discrete time. *International Journal of Production Research (3)*, 497–507

[6] Tempelmeier H, Hilger T (2015): Linear programming models for a stochastic dynamic capacitated lot sizing problem. *Computers & Operations Research (59)*, 119–125

Further Approaches and Preliminary Results on Flexible vs. Robust Lot Scheduling subject to Random Interrupted Yield and Non-Rigid Dynamic Demand

Stefan Helber
Leibniz Universität Hannover
stefan.helber@prod.uni-hannover.de

Karl Inderfurth
Otto von Guericke Universität Magdeburg
karl.inderfurth@ovgu.de

Florian Sahling
Leibniz Universität Hannover
florian.sahling@prod.uni-hannover.de

Katja Schimmelpfeng
Universität Hohenheim
katja.schimmelpfeng@uni-hohenheim.de

Abstract

We consider the problem to schedule production lots for multiple products that compete for a common production resource which processes the product units serially. The demand for each product and period is given, but the yield per production lot is random as the process can go out of control while processing each single product unit of a lot. A δ-service level constraint is used to limit the backlog in the presence of this yield uncertainty. We address the question how to schedule production lots over the discrete periods of the planning horizon. To this end, we consider different approaches ranging from stochastic dynamic programming over a flexible lot scheduling heuristic to a rigid robust planning approach. In an extensive numerical study, we compare the different approaches to assess the cost of operating according to a robust plan as opposed to a flexible policy.

A Column Generation Approach for Multiple Product Dynamic Lot-Sizing with Congestion

Yongha Kang
Quality Control Group, Samsung SDS, Republic of Korea
yyjgj@korea.ac.kr

Erinc Albey
Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, United States
ealbey@ncsu.edu

Reha Uzsoy
Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, United States
ruzsoy@ncsu.edu

Abstract

We presents a column generation (CG) approach for a multiple product dynamic lot-sizing problem on a production resource modeled as an M/G/1 queue. A master problem allocates limited capacity among n different products and sub-problems perform the lot sizing for each product subject to the capacity allocation from the master problem. The sub-problems use clearing functions that capture the non-linear relationship between expected throughput and expected WIP level in a planning period.

1 Introduction

Most deterministic lot sizing models such as the classic Wagner-Whitin model [1] focus on the trade off between the fixed cost of setup that is independent of the lot size, and the holding cost of the cycle stocks due to production occurring in batches. However, in many industrial contexts the actual cash cost of a setup is quite low; the main cost of setups is the opportunity cost of lost capacity. [2] present a multiproduct dynamic lot sizing (MDLS) model where the machine is modelled as an M/G/1 queue that captures the relationship between the expected throughput of each product as a function of the expected WIP level and lot size of all products. Computational experiments show that this model yields superior performance in terms of total costs than a model that does not consider queueing.

2 Multi-Product Dynamic Lot Sizing Model

[2] study a single machine MDLS problem where lots arrive according to a Poisson process and their processing time has a general distribution. We define the notation in Table 1, where i is the product index and t is the time period index. The lot processing time, T_{it}; waiting time T_{qt}; and utilization, ρ_{it} can be computed using the decision variables and parameters in Table 1.

Decision Variables	Parameters
R_{it}: Released quantity of product i at period t	D_{it}: Demand of product i in period t
Q_{it}: Lot size of product i at period t	h_{it}: FGI cost of product i in period t
Y_{it}: Lots of product i at period t	w_{it}: WIP cost of product i in period t
W_{it}: WIP level of product i at period t	b_{it}: BO cost of product i in period t
I_{it}: Finished goods inventory (FGI) of product i at period t	T_{it}: Lot processing time of product i in period t
B_{it}: Backorder (BO) of product i at period t	T_{qt}: Waiting time in queue in period t
	ρ_{it}: Utilization of product i in period t

Table 1: Notation.

The clearing function (CF) giving the expected output of i in t derived in [2] as:

$$f(Q_{it}, Y_{it}, \overline{\mathrm{W}}_{it}, Q_{jt}, Y_{jt}) = \frac{\overline{\mathrm{W}}_{it}}{T_{it}+\frac{Y_{it}T_{it}^2+\sum_{j\neq i} Y_{jt}T_{jt}^2}{2(1-\rho_{it}-\sum_{j\neq i}\rho_{jt})}}$$

$$= \frac{(W_{it}+W_{i,t-1})/2}{(s_i^{'}+p_i^{'}Q_{it})+\frac{Y_{it}(s_i^{'}+p_i^{'}Q_{it})^2+\sum_{j\neq i} Y_{jt}(s_j^{'}+p_j^{'}Q_{jt})^2}{2[1-Y_{it}(s_i^{'}+p_i^{'}Q_{it})-\sum_{j\neq i} Y_{it}(s_i^{'}+p_i^{'}Q_{it})]}} \tag{1}$$

where ρ_j is the utilization due to product j and T_{jp} its lot processing time; $\sum_{j\neq i}\rho_j$ is the portion of utilization allocated to products. The model (IM) of [2] is given by:

Integrated Model:IM

$$min \sum_i \sum_t h_{it}I_{it} + \sum_i \sum_t w_{it}W_{it} + \sum_i \sum_t b_{it}B_{it} \tag{2}$$

s.t

$$W_{it} = W_{it-1} + R_{it} - Q_{it}Y_{it} \ \forall i, t \tag{3}$$

$$I_{it} - B_{it} = I_{it-1} - B_{it-1} + Q_{it}Y_{it} - D_{it} \ \forall i, t \tag{4}$$

$$Q_{it}Y_{it} = f(Q_{it}, Y_{it}, \overline{\mathrm{W}}_{it}, Q_{jt}, Y_{jt}) \ \forall i, j, t \tag{5}$$

$$\sum_i Y_{it}(s_i^{'} + p_i^{'} Q_{it}) \leq 1 \ \forall t \tag{6}$$

$$Q_{it}, Y_{it} \in Z^+ \text{ and } I_{it}, W_{it}, R_{it}, B_{it} \in R^+ \ \forall i, t \tag{7}$$

The objective function (2) minimizes the sum of WIP holding, FGI holding, and backorder costs. The constraints consist of WIP balance (3), FGI inventory balance (4), the CFs (5), and the utilization constraint (6).

3 Column Generation Approach

The CG approach uses a master problem that allocates capacity among products, and a sub-problem that determines lot sizes for each product subject to this capacity allocation. Let τ_i^k denote a column vector with T entries representing a complete production schedule for product i over the periods $t = 1, ..., T$ such that each entry $\tau_{it}^k = Y_{it}^k(s_i^{'} + p_i^{'} Q_{it}^k)$ where Q_{it}^k denotes the lot size of product i in period t in schedule k and Y_{it}^k the number of lots of product i produced in period t in schedule k. Note that there are a very large number of potential schedules k. Define V_i^k as a column vector with T entries, such that $V_{it}^k = h_{it} I^k it + w_{it} W^k it + b_{it} B_{it}^k$. Let $\gamma_i^k = 1$ if schedule k is selected for product i and zero otherwise. We consider a restricted LP relaxation of the master problem, (RRMP), with a limited number of columns and $0 \leq \gamma_i^k \leq 1$, $\forall i, t$ is as follows:

Restricted Relaxed Master Problem:RRMP

$$min \sum_i \sum_k^{K_i} V_i^k \gamma_i^k \tag{8}$$

s.t

$$\sum_k^{K_i} \tau_{it}^k \gamma_i^k \leq 1 \ \forall i, t \tag{9}$$

$$\sum_k^{K_i} \gamma_i^k = 1 \ \forall i \tag{10}$$

$$0 \leq \gamma_i^k \leq 1 \ \forall i, k \tag{11}$$

Let α_{it}^k and μ_i^k be the dual variables associated with (9) and (10), respectively. The reduced cost for a new column is then $\epsilon_i = V_{it}^k + \sum_{t=1}^T \alpha_{it}^k (s_i^{'} + p_i^{'} Q_{it}^k) Y_{it}^k + \mu_i^k$ where $V_{it}^k = h_{it} I^k it + w_{it} W^k it + b_{it} B_{it}^k$. We identify an entering column by solving the sub-problem for each product i:

$$min \ V_{it}^k + \textstyle\sum_{t=1}^T \alpha_{it}^k (s_i^{'} + p_i^{'} Q_{it}^k) Y_{it}^k + \mu_i^k \tag{12}$$

s.t

$$W_{it}^k = W_{it-1}^k + R_{it}^k - Q_{it}^k Y_{it}^k \ \forall i, t \tag{13}$$

$$I_{it}^k - B_{it}^k = I_{it-1}^k - B_{it-1}^k + Q_{it}^k Y_{it} - D_{it} \ \forall i, t \tag{14}$$

$$Q_{it}^{k} Y_{it}^{k} = f(Q_{it}^{k}, Y_{it}^{k}, \overline{\mathrm{W}}_{it}^{k}, Q_{jt}^{k}, Y_{jt}^{k}) \ \forall i, j, t \tag{15}$$

$$\sum_{i} Y_{it}^{k}(s_i^{'} + p_i^{'} Q_{it}^{k}) \leq 1 \ \forall t \tag{16}$$

$$Q_{it}^{k}, Y_{it}^{k}, I_{it}^{k}, W_{it}^{k}, R_{it}^{k}, B_{it}^{k} \in R^{+} \ \forall i, t \tag{17}$$

4 Numerical Experiments

Due to the non-convexity of the feasible region, the CG solution is approximate. Our experiments consider a four product system with planning horizon of 10 periods. All products have the same costs and unit processing time, with setup costs of 20, 40, 60 and 80. The experimental design is summarized in Table 2. For each factor combination, 10 random instances are generated, yielding a total of 80 instances.

Demand mix	Demand mean	Demand coefficient of variation
0.25/0.25/0.25/0.25	100	0.1
0.10/0.20/0.30/0.40	300	0.5

Table 2: Experimental Factors and Levels.

Results indicate that the CG approach results in improvements of up to 55 % over the RIM model for some instances, although providing quite poor solutions in a limited number of instances.

5 Summary and Conclusions

We have presented a CG heuristic for the MDLS problem with queueing. Computational results show that the approach is promising. Directions for future research include examining how to further improve the computational efficiency of the CG procedures, computational testing on larger data sets, and further exploration of the implications of this analysis for cost estimation.

References

[1] Wagner, H. M. and T. M. Whitin, Dynamic Version of the Economic Lot Size Model, Management Science, 5, 89-96 (1958)

[2] Kang, Y. H., E. Albey, S. Hwang and R. Uzsoy, The Impact of Lot Sizing in Multiple Product Environments with Congestion, Journal of Manufacturing Systems, 33, 436-444 (2014)

Lot-Sizing with a Common Setup Operator and Multiple Lots - Modelling Approaches and a Solution Procedure

Karina Copil
University of Cologne, Dep. of Supply Chain Management and Production,
Albertus Magnus-Platz, D-50932 Cologne, Germany
copil@wiso.uni-koeln.de

Horst Tempelmeier
University of Cologne, Dep. of Supply Chain Management and Production,
Albertus Magnus-Platz, D-50932 Cologne, Germany
tempelmeier@wiso.uni-koeln.de

Abstract

We present modelling approaches for a dynamic capacitated lot-sizing problem with sequence-dependent setups, linked lot-sizes, multiple machines and a common setup operator. Multiple lots for the same product per macro-period are possible. A Variable Neighborhood Search procedure solves the underlying problem.

1 Introduction

We found a special lot-sizing problem in a German food company. A complex cleaning machine performs sequence-dependent setups on multiple machines. A unique assignment of products to machines exists which would usually allow the isolated solution of the simultaneous lot-sizing and scheduling problem for each machine. However, the cleaning machine can perform only one setup on one machine at a time. To take this common setup resource into account, the simultaneous lot-sizing and scheduling problem must include a synchronization of all setups to avoid overlapping

Only a small number of publications exists for this problem. [1] present a model based on the Proportional Lot-Sizing and Scheduling Problem (PLSP) for one common setup resource. [2] propose a big bucket model formulation for the CLSD (Capacitated Lot-Sizing Problem with Linked Lot-Sizes and Sequence-Dependent Setups) based on the model formulation by [3] and add a common setup resource. Numerical results show that their formulation is superior to the model by [1]. [4] present a GLSP (General Lot-Sizing and Scheduling Problem) and a CLSD formulation extended by tools which are shared by multiple machines and attached to a machine during the whole production process. As the CLSD based formulation of [2] delivers the best results and represents the problem aspects more precisely, we use this formulation as basic model for new problems.

For the CLSD formulation, it is assumed that the triangular inequality holds and only one setup and one production operation can be performed per product and period. A second setup would be associated with unnecessary costs as two lots of the same product within a period could be merged to one lot. In some cases, the triangular inequality may be violated for example if a product with a cleaning function exists and accepting multiple setups can lead to a better solution. Considering high utilizations and/or long setups in combination with an additional resource as the common setup resource can provide another reason for multiple lots. Due to the complex and limiting technical constraints, splitting production quantities and accepting additional setups can help to generate feasible production plans. [5], [6], [7] and [8] present CLSD based model formulations allowing multiple lots due to violated triangular inequalities.

However, these approaches cannot be adapted to the problem with a common setup operator because of its additional constraints. Therefore, we presented a model formulation considering a common setup operator and allowing multiple lots by using a modified time structure (see [9]). In this presentation, we propose an alternative approach to accept multiple lots. In addition, a Variable Neighborhood Search solves the underlying problem.

2 Modelling approaches

The Capacitated Lot-Sizing Problem with Sequence-Dependent Setups and Linked Lot-Sizes (CLSD) by [4] is used as basic model to handle the common setup operator. New variables documenting starting and ending times for setup operations are introduced and additional constraints coordinate the corresponding operations on each machine. An additional new binary variable defines the sequence of setups on all machines.

Two modelling approaches are used to allow multiple lots of the same product per macro-period. First, a new time structure is introduced. For each macro-period t, a set of sub-periods $\mathcal{P}_t$ is given. Considering the capacity constraints of a macro-period, each product can be produced once per sub-period. The number $|\mathcal{P}_t|$ of sub-periods per macro-period defines the maximum number of lots allowed to be produced per product in the same macro-period. Variables are modified and additional constraints coordinate operations over all sub-periods. This approach was presented by [2] and [9].

Another way to allow multiple lots is the introduction of virtual products. For each additional lot per (end-)product k, a new virtual product is introduced. The set $\mathcal{R}_k$ links the virtual products to the (end-)product k. The set $\mathcal{K}_m$ defining the product-machine association, now uses virtual products. Also, the inventory balance equations have to be modified since the demand can be fulfilled by all corresponding virtual products. This approach offers an easy way to accept multiple setups as only the data and the inventory balance equations have to be modified. The remaining model can be used in its original form. Hence, other problems and model formulations can use this simple approach without changing the underlying structure.

3 Solution approach

Both models are solved with IBM ILOG CPLEX 12.6 and the optimization language OPL. Results of the models are compared. The basic problem presented by [2] becomes more complex with increasing problem size. Enabling multiple setups complicates the underlying problem even more and solution times increase. Therefore, a General Variable Neighborhood Search using a Variable Neighborhood Descent based approach (see [10]) is developed to solve the simultaneous lot-sizing and scheduling problem with a common setup operator allowing multiple lots of the same product per macro-period. The results are compared with the exact solutions calculated by CPLEX.

References

[1] Horst Tempelmeier and Lisbeth Buschkühl, Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource. International Journal of Production Economics, 113, pp. 401–412 (2008)

[2] Horst Tempelmeier and Karina Copil, Capacitated lotsizing with parallel machines, sequence-dependent setups and a common setup operator. Working Paper, University of Cologne (2015)

[3] Ross J.W. James and Bernardo Almada-Lobo, Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. Computers and Operations Research, 38, pp. 1816–1825 (2011)

[4] Christian Almeder and Bernardo Almada-Lobo, Synchronisation of scarce resources for a parallel machine lotsizing problem. International Journal of Production Research, 49(24), pp. 7315–7335 (2011)

[5] António Aroso Menezes and Alistair Clark and Bernardo Almada-Lobo, Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. Journal of Scheduling, 14(2), pp. 209–219 (2011)

[6] Luis Guimarães and Diego Klabjan and Bernardo Almada-Lobo, Pricing, relaxing and fixing under lot sizing and scheduling. European Journal of Operational Research, 230(2), pp. 399–411 (2013)

[7] Luis Guimarães and Diego Klabjan and Bernardo Almada-Lobo, Modeling lotsizing and scheduling problems with sequence dependent setups. European Journal of Operational Research, 239(3), pp. 644–662 (2014)

[8] Alistair Clark and Masoumeh Mahdieh and Socorro Rangel, Production lot sizing and scheduling with non-triangular sequence-dependent setup times. International Journal of Production Research, 52(8), pp. 2490–2503 (2014)

[9] Karina Copil and Horst Tempelmeier, Simultaneous lotsizing and scheduling with scarce setup resources and multiple setups per period. Proceeding of the IWLS 2014 - International Workshop on Lot Sizing, pp. 46–48 (2014)

[10] Pierre Hansen and Nenad Mladenovićc and José A. Moreno Pérez, Variable neighbourhood search: methods and applications. Annals of Operations Research, 175, pp. 367-407 (2010)

A Tabu Search Heuristic for the Dynamic Capacitated Lotsizing Problem with Scarce Setup Resources

Sascha Düerkop
University of Cologne, Dep. for Supply Chain Management and Production,
dueerkop@wiso.uni-koeln.de

Dirk Briskorn
Schumpeter School of Business and Economics Wuppertal, Dep. for Production and Logistics,
briskorn@uni-wuppertal.de

Horst Tempelmeier
University of Cologne, Dep. for Supply Chain Management and Production,
tempelmeier@wiso.uni-koeln.de

Abstract

We present a new formulation for a capacitated lotsizing problem with an integrated coordination of a common setup resource, based on given schedules of the setup ressource. Furthermore, we propose a heuristic to effectively find a feasible schedule for the setup ressource. Finally we introduce a tailor-made tabu search algorithm that enhances the initial feasible schedule.

1 Introduction

A common setup operator or setup resource for a dynamic lotsizing problem was recently often found in practical cases (see for example [1] and [2]). The usual setting in industry allows setup carryovers and involves sequence dependent setup costs and times. A common setup resource requires a simultaneous lotsizing and scheduling of the setup operator.

The problem was formulated as an extension of the Capacitated Lotsizing Problem and solved by a Fix-and-Optimize (see for example [3]) approach in [1]. Even though the Fix-and-Optimize heuristic usually performs good even for large instances of the CLSP it is not a tailor-made algorithm for the particular problem. Therefore, we propose a tabu search that schedules the common setup resource efficiently.

2 Problem description of the solution approach

The overall problem is split up into two phases. In the first phase a schedule for the setup operator is determined in form of a list: Here the $l-th$ element of the list I, (i_l, t_l), symbolizes the set-up operation to product i_l, that is carried out in period t_l. This particular set-up operation is carried out on the unique machine product i_l is produced on exclusively. Thus, the predecessor of i_l is determined by I. It is the last element (i_k, t_k) with $k <= l$ and i_k is a product produced on the same machine as i_l. This information allows to determine the setup variables $x_{l,t}$ and binary order variables $\Gamma_{l,\tilde{l},t}$ as follows.

$$x_{l,t} = \begin{cases} 1 & \text{if } t = I_{l_2} \\ 0 & \text{else} \end{cases}$$

$$\Gamma_{l,\tilde{l},t} = \begin{cases} 1 & \text{if } x_{l,t} = x_{\tilde{l},t} = 1 \text{ and } l < \tilde{l} \\ 0 & \text{else} \end{cases}$$

This reduces the optimization problem to a polynomial linear problem.

3 Tabu search

3.1 Initial solution

The start heuristic determines a list of setup operations and the periods they are carried out in. During preliminary tests with practical case data the following start heuristic always found a feasible solution. However, in more general tests, it could be shown that for certain instances it does only create "‘nearly feasible”’ setup lists.

- During a set-up operation the considered machine m cannot work and any other machine cannot be set up since the set-up operator can only serve one machine at a time. Therefore, on each machine m set-up operations should consume as few time as possible. Since set-up times are sequence dependent, on each machine m the set-up time minimal set-up sequence has to be found.

- Since both the set-up operator and the machines are capacitated, not all necessary set-ups might be feasible in the period of demand. In this case, the infeasible set-ups of period t (and corresponding production) have to be shifted to an earlier period $t-n$ in order to prevent backorders. Therefore, we designed a backward oriented heuristic.

Due to the complex inter- and intra-period interdependences between sequencing and lot sizing decisions it is hard, if not impossible, to find a holistic solution procedure that creates a feasible set-up sequence. Thus the problem is decomposed into two subproblems that are solved consecutively for each period t, beginning in the last period T.

The first subproblem, $\mathcal{SUB}_m$, is to identify the set-up time minimal set-up sequence on every single machine m. Afterwards, the second subproblem, $\mathcal{SUB}_t$, generates the set-up sequence over all machines, based on the results of the M subproblems $\mathcal{SUB}_m$. $\mathcal{SUB}_m$ can be formulated as an asymmetric Travelling Salesman Problem with a weighted complete digraph where the nodes represent the products with positive demand in period t (occuring in t or deferred from a later period $t+n$), $\mathcal{V} = \{k \mid \tilde{d}_{kt} > 0,\ k \in \mathcal{K}_m\}$, and the weights assigned to the arcs, $\mathcal{E} = \{(i,k) \mid i,k \in \mathcal{V}\}$, represent the set-up times, $c\,(i,k) = tr_{ik} \quad i,k \in \mathcal{V}$. The asymmetric Travelling Salesman Problem is solved by the Fastest Insertion heuristic, which led to the best results during tests. After all $\mathcal{SUB}_m$ are solved for period t, the second subproblem $\mathcal{SUB}_t$ is tackled. Therefore, a Metra Potential Method Network consisting of a source, a sink and the M sequences obtained from the preceding step is created. The objective is to find the shortest possible project duration, i. e. performing all set-ups, under presence of a constrained resource, i. e. the set-up operator. In contrast to the classical Resource-Constrained Project Scheduling Problem (RCPSP), the maximal project duration is given by the period length, i. e. the machine capacity. Consequently, the project duration has to be minimized starting from a given endpoint. Since the RCPSP is known to be $\mathcal{NP}$-hard as well a well performing priority rule is applied: Minimum Late Finish Time (LFT).

If there is a feasible set-up sequence obtained for period t, the next period is planned. Else, a Simualted Annealing (SA) approach inspired by [4] is applied.

3.2 Neighborhood structure

Given a sequence L, we define $N(L)$ as being the set of all sequences which can be obtained from L by using one of the following 2 schemes:

1. *Swapping*: Given a sequence L, let l and $\acute{l}$ be two positions in the sequence with $L_l = (k_1, t_1)$ and $L_{\acute{l}} = (k_2, t_2)$. A neighbor of L is obtained by interchanging the products of the two setup operation, such that position l of L is changed to $L_l = (k_2, t_1)$ and position $\acute{l}$ is changed to $L_{\acute{l}} = (k_1, t_2)$. The positions l and $\acute{l}$ are selected randomly.

2. *Deletion*: Given a sequence L the l-th element of the sequence L_l is deleted. The index l is again selected randomly.

3.3 Selecting the best neighbor in the candidate list

The objective function is the minimal total of holding an setup costs. Thus, we define the sequence with the minimal resulting objective function, which is not on the tabu list, as the best neighbor. However, to allow our candidate list to be smaller and thus the algorithm to perform better, we do not undertake a full search of the list, but instead accept the first non-tabu move that improves the current solution. If there is no candidate move that improves the solution, then we consider the whole candidate list to select the best solution found.

3.4 Working with infeasible initial solutions

In case the initial solution is infeasible, the solution will be manipulated by a similar tabu search with a modified objective function. In almost all cases this results in a feasible solution which is then being used as the initial solution of the above described tabu search.

References

[1] Karina Copil and Horst Tempelmeier. *Dynamic capacitated lot-sizing with parallel common setup operators.* Working Paper, University of Cologne, 2014.

[2] Horst Tempelmeier and Lisbeth Buschkühl, *Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource.* International Journal of Production Economics, 113, pp. 401-412, 2008.

[3] Stefan Helber and Florian Sahling, *A fix-and-optimize approach for the multi-level capacitated lot sizing problem.* International Journal of Production Economic 123.2: 247-256, 2010.

[4] K. Bouleimen and H. Lecocq, *A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version.* European Journal of Operational Research 149.2: 268–281, 2003.

Designing New Heuristics for the Capacitated Lot Sizing Problem using Genetic Programming

Fanny Hein
Christian Almeder
European-University Viadrina Frankfurt (Oder)
hein@europa-uni.de
almeder@europa-uni.de

Gonçalo Figueira
Bernado Almada-Lobo
INESC-TEC, Faculty of Engineering, University of Porto
goncalo.figueira@fe.up.pt
almada.lobo@fe.up.pt

Abstract

This work addresses the well-known capacitated lot sizing problem (CLSP) which is proven to be an NP-hard optimization problem. Simple period-by-period heuristics are popular solution approaches due to the extremely low computational effort and their suitability for rolling planning horizons. The aim of this work is to apply genetic programming (GP) to automatically generate novel rules for those heuristics that establish a new state-of-the-art in the field. Preliminary tests show that we are able to obtain better solutions when using a GP-based priority rule for ranking the products, compared to the Dixon-Silver-criterion [1]. Developing heuristics for the CLSP with uncertain demand is an interesting direction for future work.

1 Introduction

We study the dynamic capacitated lot sizing problem (CLSP), a standard optimization problem and one of the most critical in production planning. Since the CLSP is NP-hard, solution methods for large instances are mostly heuristic. Simple period-by-period approaches, like Dixon-Silver [1] and ABC [4], are widely-used in research and in practice. In addition to not requiring commercial solvers, these heuristics are radically faster than mathematical programming (MP) and MP-based heuristics, which allows the approximate solution of large problem instances in real-time or their resolution as a sub-problem of a more complex problem (e.g. production routing, stochastic lot sizing). However, new constructive heuristics specific for the CLSP have not been developed since the 80s.

These heuristics comprise three main steps: (i) ranking products according to their estimated impact on the total cost; (ii) deciding if a current production lot is extended or a new lot has to be set up; (iii) a feasibility check ensuring that stocking up takes place if capacity limitations require it.

Genetic programming (GP) has been already applied to a variety of problems, but applications in operations management, except production scheduling, have not yet received attention. Our goal is to use GP to automatically generate novel decision rules to construct advanced lot sizing heuristics. We can include any problem-specific characteristics (e.g. cost ratios, current capacity utilization, demand variability) in the GP-optimization, leading to sophisticated rules that adapt to the problem environment.

2 A Genetic Programming Approach

Evolutionary computation (EC) applies the biological mechanisms of evolution to solve continuous or combinatorial optimization problems. Genetic algorithms (GAs) and GP lie at the heart of EC and, although employing the same type of evolutionary operators (e.g. reproduction, crossover, mutation), these approaches work on completely different representations. GAs [2] encode solutions in strings of 0's and 1's or real values. Each of these individuals represents a well-defined solution to the optimization problem. GP [3], on the other hand, evolves computer programs or mathematical functions which can be used to solve the problem. Solutions are represented as parse trees. The terminal nodes of the tree contain bits of information (parameters or variables) specific to the problem, whereas the internal nodes operate the former with functions (arithmetic, conditional, etc.). To provide an example, let us consider the Dixon-Silver [1] priority index u defined in (1).

$$u_i = \left(\frac{SC_i + HC_{i,T}}{T_i} - \frac{SC_i + HC_{i,T_i+1}}{T_i+1}\right)/k_i D_{i,T_i+1} \tag{1}$$

SC refers to the setup costs, HC_T and HC_{T+1} refer to the total holding cost from period 1 up to period T and $T+1$, respectively, T denotes the number of periods covered in the current lot, and kD_{T+1} expresses the additional capacity requirements needed to include the demand of period $T+1$ in the current lot. Finally, the priority index u indicates the decrease of the average costs per period per capacity unit. Figure 1 depicts u as parse tree.

In an iterative procedure, a population of expressions is transformed into a new generation of expressions by applying a combination of genetic operators. According to Koza [3], GP consists of four main executional steps:

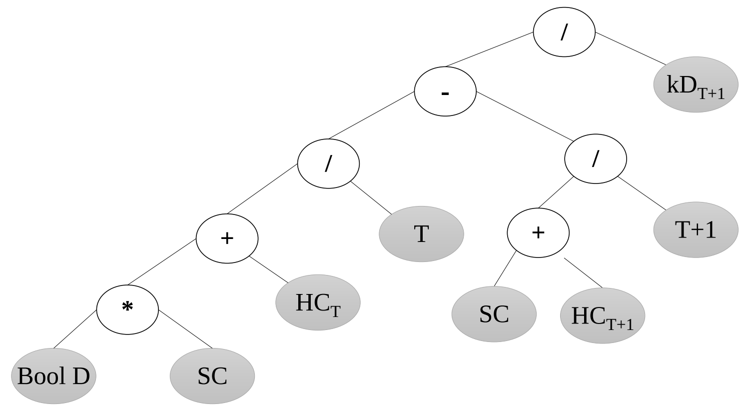

Figure 1: Dixon-Silver criterion as parse tree

Step 1: Initial population Create the initial population consisting of either randomly generated or manually defined individuals or a combination of both. In order to be able to construct individual programs, the pool of available operators and terminal nodes needs to be determined first. We consider more than 40 terminal nodes ranging from static input parameters (e.g. setup cost, capacity) to dynamic variables (e.g. cumulated remaining demand, current capacity utilization).

Step 2: Evaluate the fitness Each program is executed and evaluated according to a problem-specific fitness measure. In our tests, the fitness reflects the relative deviation between the objective value obtained by our algorithm and by CPLEX.

Step 3: Create new individuals With a probability based on its fitness value, an individual is selected to be involved in genetic operations. Steps 2 and 3 are repeated until the termination criterion is satisfied.

Step 4: Find best solution Identify the best expression found so far.

3 Preliminary Results

As a starting point, we have utilized the Dixon-Silver heuristic and replaced the priority rule for ranking the products defined in equation (1) by a GP-evolved rule. Note that the decision on lot extension further relies on index u as suggested in the original Dixon-Silver heuristic. The modified Dixon-Silver heuristic is in the following referred to as Dixon-Silver-GP.

For our preliminary tests, we used 751 test instances designed for the CLSP and introduced by Trigeiro et. al [6]. Setup times were ignored and set to zero. ABC, Dixon-Silver and Dixon-Silver-GP were implemented in C++ using Visual Studio Express 2013. For the GP-part, we used Paradiseo-EO [5], a C++ compliant evolutionary computation library. Besides, all instances were solved with CPLEX Optimization Studio 12.6. Within the limited runtime of two hours, more than 90% of the instances were solved to optimality. Table 1 denotes the relative deviation of the heuristic solution from the CPLEX solution averaged over all 751 instances.

	ABC-heuristic	Dixon-Silver	Dixon-Silver-GP
Avg. deviation from CPLEX	6.0%	7.2%	6.3%

Table 1: Average relative deviation from the CPLEX solution

Without extensive computational testing, we were able to promptly find a better priority rule than proposed by Dixon-Silver. Despite the large gap to CPLEX, we believe in the potential of GP to generate promising lot sizing rules outperforming ABC and even CPLEX for large instances. The ultimate goal is to establish rules for the CLSP with uncertain demand.

References

[1] P.S. Dixon and E.A. Silver, A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem. Journal of Operations Management, 2, 23-39 (1981)

[2] J.H. Holland, Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence, University of Michigan Press, Ann Arbor, Michigan (1975)

[3] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT press, Cambridge, Massachusetts (1992)

[4] J. Maes and L.N. Van Wassenhove, A simple heuristic for the multi item single level capacitated lotsizing problem, OR Letters, 4, 265-273 (1986)

[5] Paradiseo-EO is available at http://paradiseo.gforge.inria.fr

[6] W.W. Trigeiro, L.J. Thomas, J.O. McClain, Capacitated lotsizing with setup times, Management Science, 35, 353-366, (1989)

Dynamic Lot-Sizing-Based Working Capital Requirement Minimization Model with Infinite Capacity

Yuan Bian
IRCCyN, UMR CNRS 6597, Ecole des Mines de Nantes,
Yuan.BIAN@mines-nantes.fr

Nathalie Bostel,
David Lemoine,
Thomas Yeung,
IRCCyN, UMR CNRS 6597, Ecole des Mines de Nantes

Abstract

In this paper, we introduce a first link between tactical production planning and the financial aspects of working capital requirement (WCR). The concept of WCR is widely used in practice to assess the financial situation at a certain moment. We propose a new generic WCR model which allows us to evaluate the company's financial situation during the horizon studied. In addition, we develop a dynamic lot-sizing-based model with WCR modeling for single-site, single-level, single-product and infinite capacity cases. An exact algorithm is also presented with numerical tests in order to compare our approach with the traditional dynamic lot-sizing model.

1 Introduction

Serving the business strategy, a good production plan should make the decision of production at the right time and lowest cost. Such a plan should also include planning for the acquisition of resources, raw materials and all necessary production activities of intermediates and final products manufacturing. It naturally implies a problem which is to determine the best lot sizes to meet demand at the lowest cost, known as the lot-sizing problem. This problem aims to determine the timing and quantity of production lots with a total cost minimization objective. This total cost generally includes production, set-up and inventory cost in lot-sizing problem.

On the other hand, companies also need to cautiously manage their cash flow to ensure financial liquidity in the development phase or risk economic hardship. The working capital requirement (WCR) metric is known as a key indicator to monitor and control the financial situation of a company. We consider only the operations-related working capital requirements (OWCR) in this work. The OWCR includes all financial needs generated in the operation cycle which contains steps from receiving order to final product distribution.The OWCR is generated due to the fact that, in practice, there is a mismatch between the essential cost for production operations (i.e., accounts payable) and client payments (i.e., accounts receivable). Thus, positive OWCR represents the need for fund raising in order to cover the negative cash flow generated by the operating cycle. In contrast, a negative OWCR only appears when the operating activity generates a positive cash flow. It also represents company's financial health which ensures the capability of settling payments and a sustainable business development. In other words, a lower OWCR gives the company better liquidity and helps to ensure solvency.

2 Generic OWCR modeling

In practice, the OWCR is usually calculated by method "Accountants" and presented as normative WCR in number of days of annual revenue, see [2]. The normative WCR presents the financial need at the day of balance sheet calculation. However, we cannot track the evolution of OWCR during the considered horizon by using this method. Therefore, we propose a new generic model of the OWCR in a tactical planning context. The objective is to ensure the clear traceability of our investments in the operating cycle for tactical planning.

In order to acquire the exact amount and timing of revenues, the facility location-based formulation (FAL) is chosen as the base of our OWCR model. In this formulation, the main variable, w_{tk} represents production quantity in period t for satisfying a part of demand in period k. With these disaggregated variables, we are able to determine the total quantity produced in each period, $x_t = \sum_{k=t}^{T} w_{tk}$, and the number of periods to finance, $k - t$, (i.e., we need to finance all related cost from period t to $k - 1$). However, by following the above concept, a non-linear formula is naturally involved. The decision variables and parameters are given in Table 1.

For each w_{tk}, related production, inventory, purchasing costs can be easily written as products of unit cost, disaggregated production quantity and number of periods to be financed :

$$\text{Production cost} = pw_{tk}(k + r_c - t) \tag{1}$$

$$\text{Purchasing cost} = aw_{tk}(k + r_c - t - r_f) \tag{2}$$

$$\text{Inventory cost} = hw_{tk}(k - t) \tag{3}$$

In this model, the costs per item in production, purchasing and inventory are given as parameters. In contrast, the setup cost per product depends on different production

Parameters	Definition	Variables	Definition
n	Number of periods.	x_t	Total production quantity
d_t	External demand in period t.		in period t.
h	Unit inventory cost.	w_{tk}	Production quantity in
p	Unit production cost.		period t to satisfy the demand
q	Unit setup cost per time.		in period k ($k \geq t$).
a	Unit raw material cost.	y_t	Binary variable which indicates
r_c	Delay in payment from client.		whether indicates whether a setup
r_f	Delay in payment to supplier.		occurs in period t or not.

Table 1: Parameters and decision variables for OWCR modeling

quantities in the planning horizon; it must be calculated for each production lot. Thus, a non-linear formula of setup cost for each w_{tk} can be written as :

$$\text{Setup cost} = \frac{qy_t}{x_t + (1 - y_t)} w_{tk}(k + r_c - t) \tag{4}$$

Therefore, total OWCR (TO) consists in summing up all costs for all periods, which can be formulated as follow :

$$TO = \sum_{t=1}^{n} \sum_{k=t}^{n} [aw_{tk}(k + r_c - t - r_f) + \frac{qy_t}{x_t + (1 - y_t)} w_{tk}(k + r_c - t)$$
$$+pw_{tk}(k + r_c - t) + hw_{tk}(k - t)] \tag{5}$$

3 Dynamic LSP based model

A mixed-integer production planning model for the OWCR minimization problem, ULS_OWCR, is formulated as follows :

$$\begin{array}{llll} Min & TO & & (1.1) \\ s.t. & x_t = \sum_{k=t}^{n} w_{tk} & \forall t & (1.2) \\ & d_k y_t - w_{tk} \geq 0 & \forall t, k & (1.3) \\ & y_t - x_t \leq 0 & \forall t & (1.4) \\ & \sum_{t=1}^{k} w_{tk} = d_k & \forall k & (1.5) \\ & w_{tk} \geq 0, y_t \in \{0, 1\} & \forall t, k & (1.6) \end{array}$$

Constraints (1.2) indicate that x_t is the total quantity of production that occurs in period t. Constraints (1.3) ensure that a setup is executed before production. Constraints (1.4) prevent a setup from occurring in a period with no production. Finally, constraints (1.5) assure all demands are satisfied.

4 Resolution method and numerical tests

We have demonstrated that the property ZIO (Zero-Inventory-Ordering) remains valid for the ULS_OWCR problem in [3]. Therefore, we only examine plannings in which production would only occur when there is no product remaining in the inventory. Thus, the problem can be formulated with the acyclic oriented graph $G = \{V, E\}$. The nodes V_t represent the n periods in planning horizon with a dummy node at the end. Then, an arc E_{tk} denotes a production in period t planned to satisfy the demands between periods t and $k-1$. Therefore, the arc value represents the total cost to finance for such a production. This batching approach benefits from the ZIO property which means that production quantity can only be the sum of demands in following periods. Therefore, we avoid the difficulty caused by the nonlinear formulation of setup cost. The arc values are presented as follows :

$$
\begin{aligned}
E_{tk} = \sum_{m=t}^{k-1} [pd_m(m + r_c - t) + q(d_m / \sum_{j=t}^{k-1} d_j)(m + r_c - t) \\
+ad_m(m + r_c - t - r_f) + hd_m(m - t)]
\end{aligned}
$$

To solve the problem, a dynamic programming algorithm is established which solves the problem in O(n^2). The recursive formulation is presented as follows :

$$
TO[t] = \min_{j \in [1, t-1]} \{TO[j] + E_{jt}\}
$$

with $TO[t]$ representing the minimal total cost to finance to satisfy all demands until period t.

In order to show the interest of the ULS_OWCR model and the difference with the classical ULS model in terms of results, we compare the total OWCR obtained with the optimal solution of ULS model to the optimal OWCR obtained by ULS_OWCR model. This study is conducted on a test instance of 30 periods proposed by Trigeiro (G-72, demand 7) [4]. Overall, the average reduction in total OWCR by applying our model is approximatively 8.7%. However, a global consideration integrating the increase of logistic costs in ULS model by applying our model will be taken in account in future work.

References

[1] H. M. Wagner and T. M. Whitin, Dynamic version of the economic lot size model, *Management Science*, 5:89-96 (1958).

[2] J. C. Juhel, PhD thesis: Gestion optimale de la trésorerie des entreprises, 1978, Université de Nice.

[3] Y. Bian, N. Bostel, D. Lemoine and T. Yeung : Zero-Inventory-Ordering property in working capital requirement minimization, *Research report*, 15/03/AUTO.

[4] W. Trigeiro , L. J. Thomas, J. O. McLain : Capacitated lot-sizing with setup times, *Management Science*, 35:353-366 (1989).

Solving Two-Level Lot Sizing Problems with Inventory Bounds

Siao-Leu Phouratsamay
Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris
siao-leu.phouratsamay@lip6.fr

Safia Kedad-Sidhoum
Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris
safia.kedad-sidhoum@lip6.fr

Fanny Pascual
Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris
fanny.pascual@lip6.fr

Abstract

We study a two level Uncapacitated Lot-Sizing (2ULS) problem representing a supply chain with two actors (a supplier and a retailer) and a single product. We drive an experimental analysis to study the impact of cost transfer between the actors on their respective profit. The experimental analysis shows that it is possible to improve the supplier's profit with a holding cost transfer.

We also consider that either the supplier or retailer or both of them may have a limited storage capacity implying Inventory Bounds constraints for the 2ULS problem (2ULS-IB). We prove that with the No Lot-Splitting (NLS) constraint which means that each demand is satified by one order at the retailer level, the 2ULS-IB-NLS problems are strongly NP-hard. Finally, we propose a polynomial dynamic programming algorithm to solve the 2ULS-IB problem where the inventory bounds are at the retailer level.

1 Introduction

The optimization of the supply chain consists of determining an optimal production, transportation and distribution planning within the supply chain in order to satisfy the client demand at a minimal cost. In practice, it is difficult for the actors of the supply chain to make a decision that is optimal for the entire supply chain. Each actor of the supply chain wants to minimize his cost independently of the others. But, making their own decision based on local incentives can lead to an even worst situation for some actors of the supply chain. Thus, in order to improve the cost of the supply chain, we propose to coordinate the production and distribution planning decisions by ensuring that each actor have benefits to follow the proposed planning.

Thereafter, we consider a supply chain consisting of two actors (a supplier and a retailer) and a single product. The retailer has to satisfy a demand d_t at each period t of a planning horizon of T periods. In order to satisfy the demand, the retailer has to determine a supply plan over the horizon, *i.e.*when to order and how many units to order, that minimizes his total cost. The supplier has to determine a production plan in order to satisfy the retailer's supply plan so as to minimize his total cost. Each actor has a fixed ordering cost, a unit ordering cost and a unit holding cost. The supply chain cost is given by the sum of the supplier and the retailer costs. Minimizing the supply chain cost is equivalent to study a two-level Uncapacitated Lot-sizing (2ULS) problem where the first level is the retailer level and the second level is the supplier level.

In order to improve the cost of the supply chain, we propose to study the problem by allowing or not a cost transfer between the actors of the supply chain. Considering that the actors may have a limited storage capacity, we study the complexity of 2ULS problems with inventory bounds and demand lot-splitting constraint which imposes that each demand has to be satisfied by a unique order. Finally, we propose a polynomial algorithm to solve the 2ULS-IB$_R$ problem.

2 Coordination of the planning decisions

Li and Wang [5] propose a review about coordination mechanisms of supply chain systems in a framework that is based on supply chain decision structure and nature of demand. The majority of papers cited in [5] are applied to the Economic Ordering Quantity (EOQ) which considers a constant demand. The analysis of supply chain coordination has lead to several coordination mechanisms that are efficient in practice [2] for the EOQ problem. In our case, we consider a dynamic demand over a planning horizon of discret periods.

In order to minimize the cost of the supply chain, we have studied the impact of cost transfer between the actors on their respective profit. Minimizing the supply chain cost is equivalent to solve a 2ULS problem. We propose to coordinate the actors decisions by allowing a total cost transfer including fixed ordering cost, unit ordering cost and holding cost, or by allowing a holding cost transfer, or by considering that any cost transfer is allowed between the supplier and the retailer.

An experimental analysis on more than 100000 instances considering different cost structures show that the supplier profit can be improved when the holding cost transfer is allowed. However, transfering the holding cost implies in some cases to store the produced quantities. A limited storage can be assumed at some levels of the supply chain. Thus, we propose to analyse the complexity of 2ULS problems with inventory bounds.

3 2ULS problems with inventory bounds

We consider that either the supplier or retailer or both of them may have a limited storage capacity implying Inventory Bounds (IB) constraints for the 2ULS problem. Let us call 2ULS-IB_R the problem where the retailer has an inventory bound u^R, *i.e.*at each period $t \in \{1, \ldots, T\}$, the inventory level at the retailer has to be at most u_t^R. Similarly, the 2ULS-IB_S problem corresponds to the case where the supplier has an inventory bound u^S. Finally, 2ULS-IB_{SR} is the problem where both the supplier and the retailer have an inventory bound. We will also consider the case where both actors share the same storage facility. In this case, the total inventory level of the supplier and the retailer does not exceed the inventory bound $u^{S,R}$.

The single-level ULS problem with inventory bounds (ULS-IB) was first introduced by Love [6]. He proves that the problem with piecewize concave ordering and holding costs, inventory bounds and backlogging can be solved with an $\mathcal{O}(T^3)$ algorithm. Atamturk *et al.*[1] study the ULS-IB problem under the same cost structure assumed in Love's paper [6], considering in addition a fixed holding cost. They improve Love's algorithm and propose an $\mathcal{O}(T^2)$ algorithm to solve the problem. Recently, Hwang *et al.*[3] improve the algorithm proposed by Love [6] from $\mathcal{O}(T^3)$ to $\mathcal{O}(T^2)$ with concave costs.

A few papers deal with the two-level uncapacitated lot-sizing problem with inventory bounds. Jaruphongsa *et al.*[4] propose an $\mathcal{O}(T^3)$ algorithm to solve the 2ULS-IB_S problem with demand time window constraints. They consider that the supplier holding cost is lower than the one of the retailer and that the fixed ordering costs and the ordering costs are decreasing.

3.1 Complexity analysis

We consider at the retailer level a constraint where demand lot-splitting is not allowed, called No Lot-Splitting (NLS). This constraint implies that each demand has to be satisfied by one order. In the case where the inventory capacity is not limited, the ZIO property induces that there exists an optimal solution that satisfy the NLS constraint. We set the complexity status of some problems summarized in Table 1:

Problem	Complexity
ULS-NLS	strongly NP-hard
2ULS-IB$_R$-NLS	strongly NP-hard
2ULS-IB$_S$-NLS with demand time window	NP-hard [4]
2ULS-IB$_S$-NLS	strongly NP-hard
2ULS-IB$_{SR}$-NLS	strongly NP-hard
2ULS-IB$_{SR}$-NLS with shared inventory	strongly NP-hard

Table 1: Complexity results assuming NLS constraint

3.2 Inventory bounds at the retailer level

We propose an $\mathcal{O}(T^5)$ dynamic programming algorithm to solve the 2ULS-IB$_R$ problem. The algorithm is based on subplans decomposition induced by some structural properties of the optimal solution.

The ZIO property holds for the supplier level but does not hold for the retailer level because of the inventory bounds. We propose a block decomposition similar to the one proposed for the ULS-IB problem [1]. We denote by s_t^R the retailer inventory level at the end of the period t. We say that two periods i and j define a block denoted by $[i,j]_\beta^\alpha$ if $s_{i-1}^R = \alpha \in \{0, u_{i-1}^R\}$, $s_j^R = \beta \in \{0, u_j^R\}$ and $0 < s_t^R < u_t^R$ for all $t \in \{i, \ldots, j-1\}$. We know that the order quantity in a block $[i,j]_\beta^\alpha$ is given by $d_{ij} - \alpha + \beta$. We show that that there is at most one ordering period in every block. The principle of the algorithm is to compute the ordering period if there exists for each block of an optimal solution.

4 Conclusion

By analysing the impact of the cost tranfer between the actors on their respective profit, the retailer cost can be improved when the holding cost transfer is allowed. We prove that lot-sizing problems with NLS constraint are strongly NP-hard. Finally, we show that the 2ULS-IB_R problem is solvable in polynomial time. It can be interesting for a future work to study the 2ULS-IB_{SR} problem by considering or not that the storage is shared between the actors. We will then analyse the profit of the actors with inventory bounds when the holding cost can be transfered between the actors.

Acknowledgements This work was supported by FUI project RCMS "Risk, Credit Chain & Supply Chain Management", financed by Région Ile-de-France.

References

[1] A. Atamturk and S. Kucukyavuz. An o(n2) algorithm for lot sizing with inventory bounds and fixed costs. *Oper. Res. Lett.*, 36(3):297–299, May 2008.

[2] G. P. Cachon. Supply chain coordination with contracts, 2003.

[3] H.-C. Hwang and W. van den Heuvel. Improved algorithms for a lot-sizing problem with inventory bounds and backlogging. *Report / Econometric Institute, Erasmus University Rotterdam*, (EI 2010-17):1–30, Mar. 2010.

[4] W. Jaruphongsa, S. etinkaya, and C.-Y. Lee. Warehouse space capacity and delivery time window considerations in dynamic lot-sizing for a simple supply chain. *International Journal of Production Economics*, 92(2):169 – 180, 2004.

[5] X. Li and Q. Wang. Coordination mechanisms of supply chain systems. *European Journal of Operational Research*, 179(1):1 – 16, 2007.

[6] S. F. Love. Bounded production and inventory models with piecewise concave costs. *Management Science*, 20(3):313–318, 1973.

Robust Optimization for Lot Sizing and Scheduling Models under Uncertainty

Pedro Amorim
CEGI - INESC TEC and Faculty of Engineering of University of Porto
pamorim@fe.up.pt

Douglas Alem
Production Engineering Department, Federal University of São Carlos
douglas@ufscar.br

Eduardo Curcio
CEGI - INESC TEC and Faculty of Engineering of University of Porto
efcurcio@hotmail.com

Bernardo Almada-Lobo
CEGI - INESC TEC and Faculty of Engineering of University of Porto
almada.lobo@fe.up.pt

Abstract

The purpose of this research is to analyze the behavior, robustness and price of the robust counterpart formulation of the Generalized Lot-Sizing and Scheduling Problem with sequence-dependent setup times (GLSP) under the assumption that demands and production times are only partially known. More precisely, we assume that these parameters are bounded random variables in a given uncertainty set, but we do not know their probability distributions. The resulting problem is a mini-max formulation whose goal is to choose the best uncertainty-immunized solution. This approach is often recognized as robust optimization (RO) in the literature. Our goal is to investigate whether or not robust GLSP models produce high-quality solutions, i.e., optimal or near-optimal production and scheduling decisions at the expenses of a minor lost of performance. Moreover, it is desirable that the more operational decisions do not change too much in order to generate "stable" production plans despite the data variability. The performance of the robust counterpart formulation is assessed against three distinct stochastic programming approaches.

1 Introduction

Most of the efforts to tackle lot-sizing and scheduling problems use deterministic mixed-integer programming models [5]. In practice this means that the inventory levels that are implicitly found in these decision models are not strictly used for demand fulfillment due to demand uncertainty. Therefore, usually a stochastic inventory management model is used afterwards to tackle the uncertainty issue. There are contributions which focus on the stochastic counterpart of the lot-sizing and scheduling problem with sequence dependent setups. By incorporating explicitly uncertainty, these models end up merging the realistic planning of both lot-sizing and scheduling, and inventory management. Despite the relatively few contributions in this field, the lot-sizing and scheduling problem under uncertainty has been tackled using several different approaches [14].

Using stochastic programming, [17] address a production planning and scheduling problem, but more focused in the chemical industry. Uncertainty is tackled with a multi-stage stochastic programming model. In this case the production planning and the scheduling problems are solved in a hierarchical framework instead of integrating both problems.

[13] proposes a multi-level lot-sizing and scheduling problem with random demand that is solved using chance-constrained programming. After developing several solution methods, the computational results show the efficiency of the hybrid meta-heuristic against a exact solution algorithm and heuristics. [6] study the dynamic capacitated lot-sizing problem under random demand and a new service level measure, which reflects both the size of the backorders and waiting time of the customers. Although demand is in most of the cases the uncertainty source, there are also examples in which the stochastic lot-sizing and scheduling problem has as uncertainty source the processing time [3]. In a more recent work by [9] the uncertainty source is demand, but the processing times are controllable at a certain cost.

There are other works that assume an infinite planning horizon for a similar problem – stochastic economic lot-scheduling problem (SELSP). The literature on the SELSP is reviewed in [15] and [16]. Motivated by several practical process industries that have limited degrees of freedom when changing over between product families, [10] analyze the case of a SELSP with restrictions in the production sequence by modeling this problem as a Markov decision process. More recently, [11] relax these assumptions on the changeovers and solves this problem with a simulation-optimization approach.

Robust optimization has never been used to solve a lot-sizing and scheduling problem with sequence dependent setups. However, this approach has been used in more tactical and theoretical production planning problems. For example, [8] proposes a robust minimax model for solving a single-item lot-size problem. The effectiveness of the model is validated by numerical experiments. Moreover, the authors proved that under certain conditions the robust model converges to the more traditional stochastic model.

Our contribution addresses the gap indicated by [18] about the necessity for studying models that acknowledge the uncertain nature of operations and integrate lot sizing issues with scheduling in finite planning horizons. Notice that in other production planning problems the trend towards models that incorporate uncertainty started long before [12, 1]. In this research, we use as basis a well-known deterministic lot-sizing and scheduling model – the GLSP [4] to shed light on the differences between using a robust optimization based approach or a stochastic programming to incorporate uncertainty in processing times and demand.

2 GLSP and Uncertainty

Despite the great application in different industrial settings, the most studied GLSP assumes that input data are precisely known, or it can be reasonably approximated by expected or worst-case values, thus leading to simpler deterministic formulations. However, practical problems are subjected to both environmental and system uncertainty [7, 12]. The first case includes uncertainties that arise beyond the production process, such as demand and supply. For example, in consumer goods industry demand is commonly dependent on market factors, seasonality effects, changing customer preferences, product life cycles, and so forth. The second case includes uncertainties within the production process, such as operation yield, production and setup times, quality of raw material, etc. Uncertainty in production/setup times may arise in manufacturing plants where the production process is partly comprised by manual operations that depend on the skills of the workers, or as a consequence of machine failure.

Although stochastic programming is one of the most successful methodologies in optimization under uncertainty, this approach is sometimes criticized for being computationally prohibitive for combinatorial problems, especially for a large number of scenarios. At the same time, designing a plausible set of scenarios is often difficult due to the lack of historical data and/or to the excessive theoretical requirements for using the available scenario generation methods. These two drawbacks can be easily overcame by using RO with a polyhedral uncertainty set. Although [2] suggested that polyhedral uncertainty might result in overly conservative formulations, polyhedral sets provide tractable robust counterparts. Thus, efficient methods to solve the nominal problem may adapt well in solving the robust counterpart.

In traditional two-stage stochastic approaches, uncertainty is handled via a finite set of outcomes or scenarios in some probability space. These scenarios represent the realizations of the random variables. In this research, we consider independent realizations for production times and demands.

The flexibility to react to the uncertainty outcomes is tightened to several factors, such as the production technology/ capital insensitivity, the planning horizon and the planner attitude towards risk. In this research we extensively cover the different possibilities of reacting to uncertainty by decoupling differently the decisions to be tackled in the first and second-stage in three models.

The first model considers that both production quantities and production sequences have to be defined before uncertainty unveils. Therefore, in the second stage, the model is only able to react to the uncertainty outcomes by adjusting the demand fulfillment. This model may apply to a more conservative planner and in rigid production environments, such as the steel production planning.

The second model considers that the sequences have to defined a priori and that in the second stage the planner is able to adjust the production lots. For example, in the pulp and paper industry the production sequence for the upcoming days/weeks is fixed before the production lots as the yield of the paper machine and the adjustments of the several upstream processes is mainly dependent on it.

Finally, the third model is more in line with the traditional hierarchical planning process. Therefore, in the first stage the lot sizes are defined and their specific production sequence is left for the second stage. This case is more common in industries in which sequence dependent costs are not so crucial. For example, this model would be suitable for the production process of consumer packaged goods in which cleansing the production lines is mandatory after each production lot.

References

[1] Aloulou MA, Dolgui A, Kovalyov MY (2014) A bibliography of non-deterministic lot-sizing models. International Journal of Production Research 52(8):2293-2310

[2] Ben-Tal A, Nemirovski A (1998) Robust convex optimization. Mathematics of Operations Research 23:769-805

[3] Beraldi P, Ghiani G, Guerriero E, Grieco A (2006) Scenario-based planning for lot-sizing and scheduling with uncertain processing times. International Journal of Production Economics 101(1):140-149

[4] Fleischmann B, Meyr H (1997) The general lotsizing and scheduling problem. Operations-Research-Spektrum 19(1):11-21

[5] Guimarães L, Klabjan D, Almada-Lobo B (2014) Modeling lotsizing and scheduling problems with sequence dependent setups. European Journal of Operational Research

[6] Helber S, Sahling F, Schimmelpfeng K (2013) Dynamic capacitated lot sizing with random demand and dynamic safety stocks. OR spectrum 35(1):75-105

[7] Ho C (1989) Evaluating the impact of operating environments on mrp system nervousness. International Journal of Production Research 27:1115-1135

[8] Klabjan D, Simchi-Levi D, Song M (2013) Robust stochastic lot-sizing by means of histograms. Production and Operations Management 22(3):691-710

[9] Koca E, Yaman H, Aktürk MS (2015) Stochastic lot sizing problem with controllable processing times. Omega 53:1-10

[10] Liberopoulos G, Pandelis DG, Hatzikonstantinou O (2013) The stochastic economic lot sizing problem for non-stop multi-grade production with sequence-restricted setup changeovers. Annals of Operations Research 209(1):179-205

[11] Löhndorf N, Riel M, Minner S (2014) Simulation optimization for the stochastic economic lot scheduling problem with sequence-dependent setup times. International Journal of Production Economics

[12] Mula J, Poler R, Garcia-Sabater J, Lario F (2006) Models for production planning under uncertainty: A review. International Journal of Production Economics 103:271-285

[13] Ramezanian R, Saidi-Mehrabad M (2013) Hybrid simulated annealing and mip-based heuristics for stochastic lotsizing and scheduling problem in capacitated multi-stage production system. Applied Mathematical Modelling 37(7):5134-5147

[14] Sahinidis NV (2004) Optimization under uncertainty: state-of-the-art and opportunities. Computers & Chemical Engineering 28(6):971-983

[15] Sox CR, Jackson PL, Bowman A, Muckstadt JA (1999) A review of the stochastic lot scheduling problem. International Journal of Production Economics 62(3):181-200

[16] Winands EM, Adan IJ, Van Houtum G (2011) The stochastic economic lot scheduling problem: a survey. European Journal of Operational Research 210(1):1-9

[17] Wu D, Ierapetritou M (2007) Hierarchical approach for production planning and scheduling under uncertainty. Chemical Engineering and Processing: Process Intensification 46(11):1129-1140

[18] Zhu X, Wilhelm WE (2006) Scheduling and lot sizing with sequence-dependent setup: A literature review. IIE transactions 38(11):987-1007

Flexible Multi-Product Lot Sizing with a Static-Dynamic Uncertainty Strategy

Svenja Lagershausen
Leibniz Universität Hannover
svenja.lagershausen@prod.uni-hannover.de

Stefan Helber
Leibniz Universität Hannover
stefan.helber@prod.uni-hannover.de

Abstract

We present a stochastic single-level, multi-product dynamic lot-sizing problem subject to a production capacity constraint. The production schedule is determined such that the expected costs are minimized. The costs considered are set up and inventory holding costs as usual and additionally backlog costs and costs for overtime. The backlog is limited using a δ-service-level constraint. The expected backlog and physical inventory functions subject to the cumulated production quantity lead to a non-linear model that is approximated by a linearization approach.

1 Introduction

Lot sizing problems have been addressed in literature considering many different aspects such as period-overlapping set up times, common set up operators, perishable goods and many more. The majority of the lot sizing literature focusses on the situation in which the demand is deterministically known in advance. However, in the industrial planning procedure, the demand is usually forecasted such that deterministic values are provided.

In this paper, we address the problem of stochastic demand which is given by the expected value and the standard deviation in a stochastic capacitated lot-sizing problem (SCLSP). We consider a single machine which has a strict capacity limitation, however, a limited amount of overtime can be imposed. In this SCLSP, we assume that unmet demand can be back-ordered. The model contains a constraint on the resulting expected backlog, i.e., the cumulated yet unmet demand, to provide a specific service level to the customers. The service level used is the δ service level introduced by Helber, Sahling, and Schimmelpfeng (2013). Both the expected physical inventory and the backlog in the SCLSP are non-linear functions of the cumulated production quantity. We first provide a non-linear model formulation and then use a linearization approach such that a linear program can be used to solve the problem. The modeling approach is solved using a specific fix-and-optimize algorithm as introduced in Helber and Sahling (2010).

Bookbinder and Tan (1988) described three fundamental strategies for probabilistic lot-sizing problems subject to a service-level constraint for a single product. In the "static uncertainty" approach, both the timing and the size of production quantities are determined in advance of the demand realizations. In the other extreme it is assumed that all production decisions for each period are made when the period demands are finally known which is the "dynamic uncertainty" approach. To strike a compromise between the extremes, Bookbinder and Tan (1988) proposed the "static-dynamic uncertainty" approach in which the production periods are fixed beforehand by a given set up pattern, however the production quantities are determined after the demand has been realized. In this paper, we use an adapted version of the "static uncertainty" approach.

2 The non-linear stochastic capacitated lot sizing problem with a δ_k service level constraint

We consider the following problem:

- The objective is to find a production plan for $k = 1, \ldots, K$ products in $t = 1, \ldots, T$ periods sharing a single machine which has a time-wise limited production capacity for each period t, C_t.
- The total overtime per period over all products is allowed to the extend of O_{max} time units on the machine. Each time units has oc product-unspecific overtime costs per time period.
- The stochastic demand $D_{k,t}$ is described by the expected value $E(D)_{k,t}$ per period and product and a standard deviation σ_{kt} for each product k and time period t.
- The realization of the demand, $d_{k,t}$, is known at the beginning of the period.
- The target service level per product is δ_k.

Next to the expected value and standard deviation, the model contains the further parameters which are listed in Table 1. There are product-specific holding and set up costs for each product k denoted by hc_k and sc_k. The product specific processing times and set up times for each product k are tb_k and tr_k.

C_t	capacity in period t
$E(D)_{k,t}$	expected demand of product k in period t
$d_{k,t}$	demand realization for product k in period t
δ_k	target service level for product k
hc_k	holding cost for product k
oc	overtime costs per unit of any product in any period
O_{max}	Maximum amount of overtime per period
sc_k	set up costs for product k
$\sigma_{k,t}$	standard deviation of the demand for product k in period t
tb_k	processing time for one unit of product k
tr_k	set up time for product k

Table 1: Parameters of the mathematical model

The model is subject to the decision variables given in Table 2.

- The set up variable $X_{k,t}$ indicates whether the machine is set up for product k in period t.
- Overtime is utilized in the amount of O_t time units per period.
- The production quantity of product k in period t is $Q_{k,t}$.
- The physical inventory amounts to $YP_{k,t}$ in period t for product k. It is given by

$$YP_{k,t} = \max\left(0, \sum_{\tau=1}^{t}(Q_{k,\tau}) - d_{k,\tau}\right) \qquad \forall k, t. \tag{1}$$

- The service-level-conformal backlog level $BL_{k,t}$ is

$$BL_{k,t} = \max\left(0, \sum_{\tau=1}^{t}(d_{k,\tau} - Q_{k,\tau})\right) \qquad \forall k, t. \tag{2}$$

$X_{k,t} \in \{0;1\}$	binary set up variable that is equal to 1 if the machine is set up for product k in period t
$O_t \geq 0$	overtime in period t for all products
$Q_{k,t} \geq 0$	production quantity of product k in period t
$Y_{k,t}$	inventory of product k in period t
$YP_{k,t} \geq 0$	physical inventory of product k in period t
$BL_{k,t} \geq 0$	backlog of product k in period t allowed by the service level constraint

Table 2: Variables of the mathematical model

References

Bookbinder, J. and J.-Y. Tan (1988). Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science 34*(9), 1096–1108.

Helber, S. and F. Sahling (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problems. *International Journal of Production Economics 123*(2), 247–256.

Helber, S., F. Sahling, and K. Schimmelpfeng (2013). Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR Spectrum 35*(1), 75–105.

A New Formulation and Dynamic Cut Generation Approach for the Static-Dynamic Uncertainty Strategy

Onur A. Kilic, Huseyin Tunc, S. Armagan Tarim
Institute of Population Studies, Hacettepe University, Ankara, Turkey
onuralp,huseyin.tunc,armagan.tarim@hacettepe.edu.tr

Roberto Rossi
Business School, University of Edinburgh, Edinburgh, United Kingdom
roberto.rossi@ed.ac.uk

Abstract

In this paper, we present an extended MIP formulation of the stochastic lot-sizing problem for the static-dynamic uncertainty strategy. The proposed formulation is significantly more time-efficient as compared to existing formulations in the literature and it can handle variants of the stochastic lot-sizing problem characterized by penalty costs and service level constraints, as well as backorders and lost sales. Also, besides being capable of working with a predefined piecewise linear approximation of the cost function – as is the case in earlier formulations, it has the functionality of finding an optimal cost solution with an arbitrary level of precision by means of a novel dynamic cut generation approach.

1 Background

The lot-sizing problem aims at determining a minimum cost inventory plan so as to meet demand over a finite discrete planning horizon. The lot-sizing problem and its variants are traditionally studied under the assumption of deterministic demands. However, there is a growing body of work on more realistic lot-sizing problems where demands are assumed to be random variables. This assumption has a significant impact on lot-sizing problems since the inventory position as well as the costs incurred in later periods now become random variables following random demands. As a result, if demands are random, then the solution to a lot-sizing problem is no longer a deterministic inventory plan but an inventory policy. This policy defines, for any given period and inventory position, whether to order, and if so how much to order. Bookbinder and Tan [1] provide a broad classification of such policies that can be employed in stochastic lot-sizing problems. Among these, is the so-called static-dynamic uncertainty strategy. Following this strategy, one sets the number as well as the timing of all orders at the very beginning of the planning horizon, and then, at each replenishment epoch decides the order quantity upon observing the inventory position. The static-dynamic uncertainty is an appealing strategy since it eases the coordination between supply chain players [2, 7], and facilitates managing joint replenishments [5] and shipment consolidations [3]. As such, an expanding line of research has been emerged on computational methods for the static-dynamic uncertainty strategy.

2 Contribution

An important direction of research is to develop computationally efficient formulations of the stochastic lot-sizing problem under the static-dynamic uncertainty strategy. Tunc et al. [8] recently proposed such a MIP formulation that make use of the network flow structure of the problem. This formulation has a tighter linear relaxation as compared to earlier formulations, and in turn it has a superior computational performance. However, their formulation is designed solely for problems characterized by α service levels. The work we carry out in the current paper extends their approach so as to capture more general variants of the problem. We contribute to the literature by presenting a new MIP formulation of the stochastic lot-sizing problem for the static-dynamic uncertainty strategy. The proposed formulation is significantly more time-efficient as compared to existing formulations in the literature and it can handle variants of the stochastic lot-sizing problem characterized by penalty costs and service level constraints, as well as backorders and lost sales. Also, besides being capable of working with a pre-defined piecewise linear approximation of the cost function – as is the case in earlier formulations, it can find a minimum cost solution with an arbitrary level of precision by means of a novel dynamic cut generation approach.

3 Computational Results

The purpose of the computational study is the demonstrate the efficiency of the extended formulation with and without the dynamic cut generation approach. We are interested, in particular, to analyze how the extended formulation performs as compared to existing formulations in the literature and whether the dynamic cut generation approach scales well.

In the first part of the computational study, we assess the computational performance of the extended formulation against a benchmark formulation employed by Tarim and Kingsman [6] and Rossi et al. [4]. We abbreviate the extended formulation and the benchmark formulation as PM and BM, respectively.

Table 1 presents the results of the computational study for different planning horizon lengths. Here, S-GAP is the % integrality gap, E-GAP is the % optimality gap at termination (with a time limit of half an hour), TIME is the solution time in seconds, and NODES is the number of explored nodes. The results demonstrate without doubt that PM is more time efficient than BM. This finding is consistent under all parameter settings. PM solves all instances to optimality in 1.31 seconds on average. BM, on the other hand, fails to solve 1 out of 10 instances within the time limit, and averages a solution time of 261.35 seconds. The dominance of PM stems from its tight linear relaxation. PM averages an S-GAP of 0.11% and a NODES of 0.09. As such, most of the time an optimal solution can be found at the root node.

		BM				PM			
		S-GAP	E-GAP	TIME	NODES	S-GAP	E-GAP	TIME	NODES
	20	52.18	0.00	0.69	$3.6\,10^3$	0.00	0.00	0.24	0.02
N	30	62.58	0.01	55.85	$2.1\,10^5$	0.01	0.00	0.95	0.13
	40	68.25	6.93	727.50	$1.1\,10^6$	0.01	0.00	2.73	0.12
Average		61.00	2.31	261.35	$4.2\,10^5$	0.01	0.00	1.31	0.09

Table 1: The solution statistics of BM and PM

In the second part of the computational study, we conduct further experiments to analyze how the dynamic cut generation approach scales as compared to the stand-alone extended formulation. We will abbreviate the dynamic cut generation approach as Cuts. Table 2 presents the average computation times of PM and Cuts in seconds for planning horizon lengths up to 120 periods. The results show that both PM and Cuts solve all problem instances to optimality in less than quarter of an hour. Thus, it is fair to say that both methods can solve realistic-sized instances to optimality in reasonable computational times. We also observe that Cuts is around two times faster than PM in finding an optimal solution. This highlights that one can bypass the off-line piecewise linearization of the cost function and establish an even more precise approximation on-the-fly without sacrificing computational efficiency. The importance of this result becomes more evident if we recall that finding a good piecewise linear approximation is a challenging optimization problem itself.

N	50	60	70	80	90	100	110	120
PM	9.09	25.77	47.51	84.09	192.80	298.06	519.73	865.71
Cuts	6.14	14.69	22.86	38.29	108.57	123.61	248.80	431.41

Table 2: The computational times of PM and Cuts

References

[1] J. H. Bookbinder and J. Y. Tan. Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, 34(9):1096–1108, 1988.

[2] O. A. Kilic and S. A. Tarim. An investigation of setup instability in non-stationary stochastic inventory systems. *International Journal of Production Economics*, 133 (1):286–292, 2011.

[3] F. Mutlu, S. Cetinkaya, and J. H. Bookbinder. An analytical model for computing the optimal time-and-quantity-based policy for consolidated shipments. *IIE Transactions*, 42:367–377, 2010.

[4] Roberto Rossi, Onur A. Kilic, and S. Armagan Tarim. Piecewise linear approximations for the static–dynamic uncertainty strategy in stochastic lot-sizing. *Omega*, 50:126–140, 2015.

[5] E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling*. Wiley New York, 1998.

[6] S. A. Tarim and B. G. Kingsman. Modelling and computing (R^n, S^n) policies for inventory systems with non-stationary stochastic demand. *European Journal of Operational Research*, 174:581–599, 2006.

[7] H. Tunc, O. A. Kilic, S. A. Tarim, and B. Eksioglu. A simple approach for assessing the cost of system nervousness. *International Journal of Production Economics*, 141(2):619–625, 2013.

[8] Huseyin Tunc, Onur A Kilic, S Armagan Tarim, and Burak Eksioglu. A reformulation for the stochastic lot sizing problem with service-level constraints. *Operations Research Letters*, 42(2):161–165, 2014.

A Lot-Sizing and Bin-Packing Heuristic for Outbound Shipment Consolidation

Tiffany A. Bayley
Department of Management Sciences, University of Waterloo, Canada
tiffany.bayley@uwaterloo.ca

James H. Bookbinder
Department of Management Sciences, University of Waterloo, Canada
jbookbinder@uwaterloo.ca

Abstract

Shipment consolidation is typically performed according to the origin of orders, and once the assignment of orders to containers is made, it may remain the same over the planning horizon. However, consideration of order due dates is also important. We iterate between a bin-packing problem that determines order-container assignments, and a joint replenishment problem that specifies the size and shipment date of each order. Our heuristic allows order-container assignments to change over time so that container utilization is maximized and ordering and shipment costs are lowered.

1 Introduction

We consider a shipment consolidation problem where order consolidation is done proactively at an earlier stage of the replenishment cycle, as in Crainic et al. [2]. If consolidation strategies are determined based on forecasted demand, it is possible to consider each container as a "family" of orders. Once orders are grouped into families, the volume of each order to ship can be determined. This inbound consolidation problem can then be viewed as a lot-sizing problem, specifically a multi-family joint replenishment problem: Given the assignment of orders to containers, the demand of each order in every period (expressed in volume), and the capacity of all the containers, find the volume of each order to ship in any period that will minimize the cost of order setups plus holding costs. To the best of our knowledge, only Erenguc and Mercan [3, 4] study the multi-family version of the joint replenishment problem.

The bin-packing problem (BPP) and multi-family joint replenishment (MFJRP) share characteristics, where the output of one model can be used as the input for the other. Ben-Khedher and Yano [1] solve a sequential joint replenishment problem

(JRP) and BPP to determine the optimal delivery schedule of orders that minimizes the number of truckloads. The JRP provides the lowest-cost delivery schedule to meet demand, while the BPP optimizes the loading configuration. Remaining capacity in each truck is computed, and along with orders that have not yet been assigned to trucks, a single-item lot-sizing problem is solved to maximize container utilization.

Unless demands are constant, order due dates will be time-varying. If we allow orders to be grouped based on their due dates, this will influence the way in which order families (containers) are defined. Rather than force order-container assignments to remain static over a planning horizon, we show how *dynamic family assignment* can help carriers and third-party logistics providers achieve greater cost savings, either through reduced transportation or inventory holding costs. We adopt the framework of [1] in the context of our problem. In each period, a BPP is solved to provide order-container assignments along with order-volume to ship. The MFJRP then "freezes" these order-container assignments over the planning horizon and treats the shipment sizes as forecasted demand.

2 Problem Formulation

A firm serves multiple customer zones, shipping to each zone numerous orders that are sourced from one common geographical region. Customer orders (demands) are known for each order $i = 1, 2, \ldots, m$ in every month $s = 1, 2, \ldots, S$, and are expressed in terms of volume. The firm must determine which orders to fulfill each week along with the lowest-cost consolidation plan; this means that the firm must communicate to its suppliers the orders it wishes them to send, and must specify how to group the orders so that customer demands are met while not exceeding container capacities.

Monthly demand can be viewed as an aggregate of several weeks' demand and should be more accurate to forecast. This allows for some flexibility in the weekly shipment of material: as long as orders arrive within the month they are required, they will not need to be backordered (though inventory holding costs will still apply).

Since orders are shipped on a weekly basis, weekly demand is determined from the monthly demand. We let D_{is} be the monthly demand of item i in month s. Weekly demand is then generated by applying an appropriate probability distribution, resulting in $\overline{d}_{it}$, demand for order i in period $t = 1, 2, \ldots, n$.

These demands are used in a *dynamic family assignment* (DFA) heuristic, which consists of two main routines: (i) a bin-packing problem to assign orders to containers; and (ii) a multi-family joint replenishment problem to determine shipment volumes in each period. Each routine will now be discussed in further detail.

2.1 Bin-Packing Problem

Based on the given weekly demand (determined from the monthly forecasted demand), we solve a bin-packing problem $[BPP_t]$ for every t. Orders may be divided across multiple containers. For orders shipped in full containers (i.e. all container capacity is used), we consider those demands to be satisfied and fix those shipment quantities. The remaining demand in each period can then be calculated.

From the partially-filled containers, we determine a fixed assignment of orders to containers based on the maximum volume shipped over all containers and time periods from the $[BPP_t]$. Remaining demand to be shipped is expressed as d_{ijt}, the volume of order i to be shipped in container j in period t. These assignments and demands are used in the $[MFJRP]$ heuristic, discussed next.

2.2 MFJRP Heuristic

Each partially-filled container can be viewed as a family, and we must determine in which periods to ship the remaining required demand. The MFJRP is formulated as a transportation-type problem (as in [5, 6]), where we minimize the total costs of major and minor setups and the associated unit cost of shipping, holding, and "penalties". Earlier, we noted that as long as orders are delivered within the month, backorders will not be needed. Here, we introduce backlogging penalties to discourage shipments from being postponed too far into the month, but to also allow for flexibility in the schedule. Constraints ensure that demand is met, capacity is not exceeded, and that shipments are dispatched only if the proper order and family setups have occurred.

$[MFJRP]$ is solved using a Lagrangian heuristic, combined with a genetic algorithm enhanced with variable neighbourhood search. In each period, we will have obtained the volume of order i to ship in container j, which can be added to the quantity of fixed orders shipped that was determined from $[BPP_t]$. An updated value of $\overline{d}_{it}$ is calculated, and we solve $[BPP_t]$ again for each period t. Order container-assignments may be different now due to the updated demand values, and may affect container utilization.

The DFA heuristic iterates between $[BPP_t]$ and $[MFJRP]$ until order-container assignments stabilize.

3 Solution Approach

Computational experiments are derived from data sets used in Trigeiro et al. [7]. A full factorial analysis is conducted using the number of orders $m = 4, 6, 8, 10$, and number of months $S = 3, 6, 8$. We allow the number of periods within a month to vary, giving $n = 2S, 4S$ (i.e. either 2 or 4 periods each month). We generate five different instances for each problem size, for a total of 120 test instances.

The performance of the DFA heuristic is measured against the Ben-Khedher and Yano methodology [1] (when multiple families are not considered in their joint replenishment problem). As well, we compare our DFA results to the case where the bin-packing problem alone determines how orders are consolidated and shipped.

References

[1] N. Ben-Khedher and C.A. Yano. The multi-item joint replenishment problem with transportation and container effects. *Transportation Science*, 28(1), 37–54 (1994).

[2] T.G. Crainic, S. Marcotte, W. Rei, and P.M. Takouda. Proactive order consolidation in global sourcing. In J.H. Bookbinder (Ed.), *Handbook of Global Logistics*, 501–530. Springer, New York (2013).

[3] S.S. Erenguc and H.M. Mercan. A multifamily dynamic lot-sizing model with coordinated replenishments. *Naval Research Logistics*, 37(4), 539–558 (1990).

[4] H.M. Mercan and S.S. Ereguc. A multi-family dynamic lot-sizing problem with coordinated replenishments: a heuristic procedure. *International Journal of Production Research*, 31(1), 173–189 (1993).

[5] O. Solyalı and H. Süral. The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. *Annals of Operations Research*, 196(1), 517–541 (2012).

[6] H. Süral, M. Denizel, and L.N. Van Wassenhove. Lagrangean relaxation based heuristics for lot sizing with setup times. *European Journal of Operational Research*, 194(1), 51–63 (2009).

[7] W.W. Trigeiro, L.J. Thomas, and J.O. McClain. Capacitated lot sizing with setup times. *Management Science*, 35(3), 353–366 (1989).

Fix-and-Optimize Heuristic and MP-based Approaches for Capacitated Lot Sizing Problem with Setup Carryover, Setup Splitting and Backlogging

Cheng-Lung Chen
School of Computing, Informatics & Decision System Engineering, Arizona State University
chenglung@asu.edu

Srimathy Mohan
W.P. Carey School of Business, Arizona State University
srimathy@asu.edu

Muhong Zhang
School of Computing, Informatics & Decision System Engineering, Arizona State University
mzhang21@asu.edu

Abstract

In this paper, we focus on the single-level, multi-item capacitated lot sizing problem with setup carryover, setup splitting and backlogging. Although the capacitated lot sizing problems have been investigated with many different features from researchers, the simultaneous consideration of setup carryover and setup splitting is relatively new. This consideration is beneficial to reduce costs and produce feasible production schedule. In this research, we use a simple plant location reformulation of the original mixed integer programming model to obtain a tighter formulation. We also add valid inequalities to further tighten the formulation. A fix-and-optimize heuristic with two-stage product decomposition and period decomposition strategies is proposed to solve the formulation. The computational results show the capability, flexibility and effectiveness of the approaches, achieving 6% and 8% average optimality gap for data without and with backlogging, respectively.

1 Introduction

One of the recent research trends on big bucket capacitated lot sizing problem (CLSP) is to include setup carryover, setup splitting and backlogging. Considering setup splitting in the formulation either generates a better production schedule in terms of lower costs or removes infeasibility from models with setup carryover only [4]. A possible condition for infeasibility is when the length of setup time is substantially long and might even surpasses the capacity of the time period. Long setup time is ubiquitous in some process manufacturing industries and automobile production processes. Therefore it is necessary to take setup splitting into account in order to have a feasible production schedule. In this research, we first study the single-level, multi-item, capacitated lot sizing problem with setup carryover and setup splitting (CLSP-SCSS) and then incorporate demand backlogging in the model. As an extension of the basic CLSP, the CLSP-SCSS is also NP-hard. It is unlikely to solve the problem within reasonable time limit by traditional exact methods such as branch-and-bound when problem size is getting larger. To address this problem, the goal of this research is to focus on developing an efficient solution procedure for the CLSP-SCSS.

2 Solution Approach

The CLSP-SCSS is reformulated by using simple plant location formulation (SPL-SCSS) to generate a tighter lower bound and formulation such that computational burden can be reduced. Three types of valid inequalities [1]: pre-processing inequalities, inventory/setup inequalities and single-item production inequalities are added to the model at the beginning. Then a fix-and-optimize heuristic [2, 3] is adopted to solve the proposed formulation. This generic heuristic solves a small portion of binary variables, that is, the binary setup state variables and all the other continuous decision variables are optimized together rapidly in each subproblem. For constructing the initial solution, we also used rounding heuristic to round up all the fractional setup variables after LP relaxation and obtain corresponding setup carryover and setup splitting to generate an initial setup pattern. In some small size problems, this initialization with the fix-and-optimize heuristic could reach the true optimal solutions. In addition, we incorporated the case with demand backlogging to demonstrate that making additional assumptions to the model does not require to completely altering this heuristic.

3 Computational Result

We took the data used in [5] to test the capability of the proposed solution method. Each test data set has 270 data instances with number of products range from 5 to 15 and the length of the planning horizon ranges from 20 to 40 periods in our test problems. In each data set, the average setup time of all items is fixed to 40%, 70% or 120% of period capacity (400, 700 and 1200 since period capacity is 1000). The proposed formulation, valied inequalities, rounding heuristic and fix-and-optimize heuristic are coded by AMPL and solved by IBM ILOG CPLEX 12.0.6.1. A Dell Precision T7500 Workstations is used to perform all the computational test. The processor of the workstation is dual six-core Intel Xeon Processor X5690 (4.46 GHz, 12M L3, 6.4 GT=s). The memory of the workstation is 48 GB, 1,333 MHz, DDR3RDIMM, ECC (6DIMMS).

Here we give the result summary of data without backlogging in Table 1, while the result of data with backlogging is presented in Table 2. From the experiment result, setup splitting variables is essential for finding feasible solution when long setup time exists and problem size become bigger. When backlogging is not allowed, the performance of algorithm tends to be stable in terms of average optimality gap regardless of the number of periods. This property can be beneficial for company to extend the planning horizon, which is more common and practical in reality compared to increase product types. The result of data with backlogging is in general harder to solve, yet the fix-and-optimize heuristic still provides competitive results in terms of optimality gap.

	Time (s)	Gap (%)
Maximum	439.8032	35.43
Average	73.555	5.56
Product		
5	30.56	4.17
10	69.525	5.93
15	120.58	6.58
Period		
20	37.397	5.23
30	64.473	5.54
40	118.79	5.91
Setup Time		
40	51.186	1.91
70	69.255	3.63
120	100.22	11.14

Table 1: Summary of Optimality Gap and Time for SPL-SCSS

	Time (s)	Gap (%)
Maximum	5473.788	29.38
Average	135.782	8.00
Product		
5	32.75	6.26
10	98.61	7.99
15	275.98	9.74
Period		
20	56.67	7.22
30	101.37	8.31
40	249.30	8.46
Setup Time		
40	73.06	
70	112.30	7.98
120	221.99	13.30

Table 2: Summary of Optimality Gap and Time for SPL-SCSS-BL

4 Conclusion

Modeling setup features in lot sizing problems is usually using binary variables. Adopting fix-and-optimize heuristic to solve the problem is advantageous as the heuristic mainly decomposes binary variables, resulting in smaller subproblems. We also demonstrate the effectiveness of combining other mathematical programming based approaches including rounding heuristic, reformulation as well as valid inequalities. Another advantage of fix-and-optimize heuristic would be the easiness to implement and flexibility when other features are included in the model, such as backlogging in our case. A comprehensive evaluation with more instances and other potential algorithms would be beneficial. We are currently extend this model with parallel parallel machines as well as multi-level production structures.

References

[1] Suerie, C., & Stadtler, H. The capacitated lot-sizing problem with linked lot sizes. Management Science, 49(8), 1039-1054 (2003)

[2] Sahling, F., Buschkühl, L., Tempelmeier, H., & Helber, S. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. Computers & Operations Research, 36(9), 2546-2553 (2009)

[3] Helber, S., & Sahling, F. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. International Journal of Production Economics, 123(2), 247-256 (2010)

[4] Mohan, S., Gopalakrishnan, M., Marathe, R., & Rajan, A. A note on modelling the capacitated lot-sizing problem with set-up carryover and set-up splitting. International Journal of Production Research, 50(19), 5538-5543 (2012)

[5] Belo-Filho, M. A., Toledo, F. M., & Almada-Lobo, B. Models for capacitated lot-sizing problem with backlogging, setup carryover and crossover. Journal of the Operational Research Society, 65(11), 1735-1747 (2013)

Fix and Relax and Rolling Horizon Algorithms for Profit maximization General Lot sizing and Scheduling Problem (PGLSP)

Narges Sereshti
Graduate student, Department of Industrial Engineering, Dalhousie University
narges.sereshti@dal.ca

Mehdi Bijari
Associate professor, Department of Industrial Engineering, Isfahan University of Technology
Bijari@cc.iut.ac.ir

Abstract

In this research, Profit maximization General Lot sizing and Scheduling Problem with demand choice flexibility (PGLSP) is studied. To solve this problem, different heuristic algorithms based on proposed mathematical models are presented and compared against the exact method. These algorithms are classified into two categories, Rolling Horizon, and Fix and Relax. While these algorithms reduce the execution time reasonably, their solutions have a good quality.

1 Introduction

Lot sizing and scheduling are two important problems in the field of production planning. Despite the fact that these problems are dependent to one another, in most researches they are analyzed separately and hierarchically [1]. Considering the interactions, the General Lot sizing and Scheduling Problem (GLSP) considers these two issues as one problem [2] .

The basic assumption in most researches in the field of lot sizing and scheduling and specifically in GLSP models is that companies should respond to all predetermined demands. However in a business with the goal of maximizing benefits, fulfilling all demands may not be the best solution. In this research, Profit maximization General Lot sizing and Scheduling Problem with demand choice flexibility (PGLSP) is studied. This problem is an extension of the GLSP by adding demand choice flexibility. In other words, the amount of demand accepted in each period, lot sizing, and scheduling are problems which are considered simultaneously. Accepted demand is between the upper and lower bounds in each period. With regard to this assumption, the traditional objective function of the model, i.e., minimizing costs is changed to maximizing net profit.

PGLSP is an NP-Hard problem and the execution time increases quadratically with increase in the size of the problem [3]. In this research, different heuristic algorithms based on mathematical models are proposed. These algorithms are classified into two categories, Rolling Horizon, and Fix and Relax. The main focus of this research is on these algorithms and their comparisons.

2 Problem Definitions and Assumptions

Profit maximization in General Lot sizing and Scheduling Problem (PGLSP) is an extension to GLSP problem by adding flexibility in choosing demand. PGLSP is described as follows: Having P products and T planning periods, the decision maker wants to define: (1) the accepted demand of each product in each period which is between upper and lower bounds, (2) the quantity of lots for each product, and (3) the sequence of lots. The objective function is maximizing the revenue of sales minus production, holding, and setup costs. Assumptions of the model are as follows:

- Backlog is not allowed;
- Setup times and costs are sequence-dependent. The triangular inequality holds between setup times;
- The model has the characteristic of setup preservation, which means that if we have an idle time, setup state will not change after it;
- The breakdown of setup time between two periods is not allowed and the setup is finished in the same period in which it begins" [3].

3 Mathematical Models

Four mathematical models have been proposed for this problem [4]. These models are different in the method of lot sizing and scheduling. The first two models use micro periods for product sequencing [5]. These micro periods do not exist in the third and fourth models, which consider the sequencing problem as a Traveling Salesman Problem (TSP) model [6]. Lots in the first and third models are determined by the unit of products which are produced in each period, while in the second and fourth models they are separated by the period in which they will be used. Based on the experimental results two of the methods which use TSP formulation for scheduling have superior performance to the other two models which use small buckets [4].

4 Heuristic Algorithms

Heuristic methods based on mathematical models. Based on the two superior mathematical models seven different methods are presented. These methods are based on Rolling Horizon and Fix and Relax algorithm.

4.1 Fix and Relax (FR)

The number of binary variables in Mixed Integer Problems (MIP) has a considerable impact on solution time. The FR algorithm is an iterative procedure which breaks down the original MIP to smaller sized MIP to reach the near optimal solution in a reasonable time [7]. In this research, the break-down in done based on the planning periods. For example, in the first iteration only the original binary variables in the first period remain binary and the rest of variables are relaxed. In the next iteration, the binary variables in the first period are fixed based on the solution, in the second period defined as binary variables, and in the next periods they remain relaxed. This process will continue until the last period.

4.2 Rolling Horizon (RH)

In many companies, planning is done based on a rolling horizon as they need to adjust their plans to new demands and prices and only the first period of planning is implemented [8]. In addition, the optimal solution of the problem based on predicted data is not an optimal solution with the real data. The RH algorithm is based on this concept. Similar to FR algorithm, three sections can be defined based on the planning periods.

- The first section includes n-1 periods in which some of the decision variables are fixed based on previous iterations.
- The second section is the n^{th} period in which the problem is considered completely.
- The last section is from (n+1)th period to the end of the planning horizon which includes a simpler form of the problem. In PGLSP the simple form only considers lot sizing and there are no binary variables for scheduling [9].

The original PGLSP formulation is modified. Similar to the FR algorithm, this algorithm is an iterative procedure. In the first iteration, the modified mathematical model is solved for n=1. In the second iteration, the binary variables are fixed based on the first iteration and the modified model is solved for n=2. This process will continue until n is equal to the last period.

In this research different heuristic methods based on RH and FR are presented and their efficiencies are investigated against each other as well as the exact mathematical models.

References

[1] D. Quadt, Lot-Sizing and Scheduling for Flexible Flow Lines,Springer Verlag,New york, (2004)

[2] B. Fleischmann, H. Meyr, The general lotsizing and scheduling problem, OR Spectrum, Vol. 19, pp. 11–21 , (1997)

[3] N. Sereshti, M. Bijari, Profit maximization in simultaneous lot-sizing and scheduling problem, Applied Mathematical Modelling, Vol 37, pp. 9516–9523 (2013)

[4] N. Sereshti, Profit Maximization General Lot-sizing and Scheduling Problem with Demand Choice Flexibility, Isfahan University of Technology, Iran (2010)

[5] H. Meyr, Simultaneous lotsizing and scheduling by combining local search with dual reoptimization, European Journal of Operations Research, Vol. 120, pp. 311–326, (2000)

[6] B. Almada-Lobo, D. Klabjan, M.A. Carravilla, J.F. Oliveira, Single machine multi-product capacitated lot sizing with sequence-dependent setups, International Journal of Production Research, Vol.45, pp. 4873–4894 (2007)

[7] L.A. Wolsey, Integer programming, IIE Transactions, Vol. 32, pp. 273–285, (2000)

[8] W. Van den Heuvel, A.P.M. Wagelmans, A comparison of methods for lot-sizing in a rolling horizon environment, Operations Research Letters, Vol. 33, No. 5, pp. 486–496, (2005)

[9] Merc, C., and Fontan, G., MIP-based heuristics for capacitated lotsizing problems, International Journal of Production Economics, Vol. 85, No. 1, pp. 97–111, (2003)

Integrated Procurement and Reprocessing Planning of Perishable and Re-Usable Medical Devices in Hospitals

Steffen Rickers
Leibniz University Hannover
steffen.rickers@prod.uni-hannover.de

Florian Sahling
Leibniz University Hannover
florian.sahling@prod.uni-hannover.de

Abstract

We present a new model formulation for a multi-product economic order quantity problem with product returns and reprocessing option (OORPP, Optimal Order and Reprocessing Planning Problem). The optimization comprises the limited shelf life of sterile medical devices as well as capacity constraints of reprocessing and sterilization resources. The time-varying demand is known in advance and must be satisfied by purchasing new medical devices and/or by reprocessing used and expired ones. The objective is to determine a feasible procurement and reprocessing schedule that minimizes the incurring costs. For the solution of the OORPP, a heuristical approach based on column generation is proposed and first numerical results are presented.

1 Introduction

Because of rising costs and competitive pressure, hospital managers are often forced to develop strategies for process optimization and cost reduction. One possibility for cost reduction can be identified within non-medical, so-called secondary processes.

The preparation of hospitals with sterile, reusable medical devices represents one of these secondary processes. This process is particularly economically relevant, since this process is related to a substantial portion of total costs. Since medical devices are not manufactured in hospitals, procurement processes are necessary. Large hospitals often possess specialized reprocessing and sterilization sections that enable the re-usage of medical devices. This reprocessing option results in a material cycle in which returns of medical devices from the operating theaters and wards must be considered. Obviously, decisions concerning the timing of placing orders and of reprocessing medical devices interact directly. Thus, it is necessary to consider procurement and reprocessing activities simultaneously. An additional challenge represents the limited shelf life of sterile medical devices.

The reprocessing cycle is identical for all re-usable medical devices and can be described as follows: After using the medical devices in the operating theater, they are pre-cleaned to remove major contaminations. Additionally, a pre-screening of medical devices with uneven surfaces is necessary because contaminations are difficult to remove. Damaged medical devices are disposed of and thus leave the reprocessing cycle directly. The subsequent decontamination includes the cleaning, disinfection, rinsing and drying. Afterwards, the cleaning results are controlled and the functionality is tested. All medical devices are packed into surgical- or department-specific sets. These sets are then sterilized in order to kill remaining microorganisms. Different time-temperature combinations can be used to sterilize the very same medical devices. Since the sterilization resource represents the bottleneck of the overall process, we merely focus on its capacity since the remaining resources for reprocessing are not scarce. If the medical devices are not provided directly in operating theaters, they can be stored. However, the shelf life of sterilized medical devices depends on the type of package and storage conditions.

2 Model assumptions

In the OORPP, the planning horizon is devided into T periods ($t = 1, \ldots, T$). K different medical devices ($k = 1, \ldots, K$) can be ordered or reprocessed. The product flow is illustrated in Figure 1.

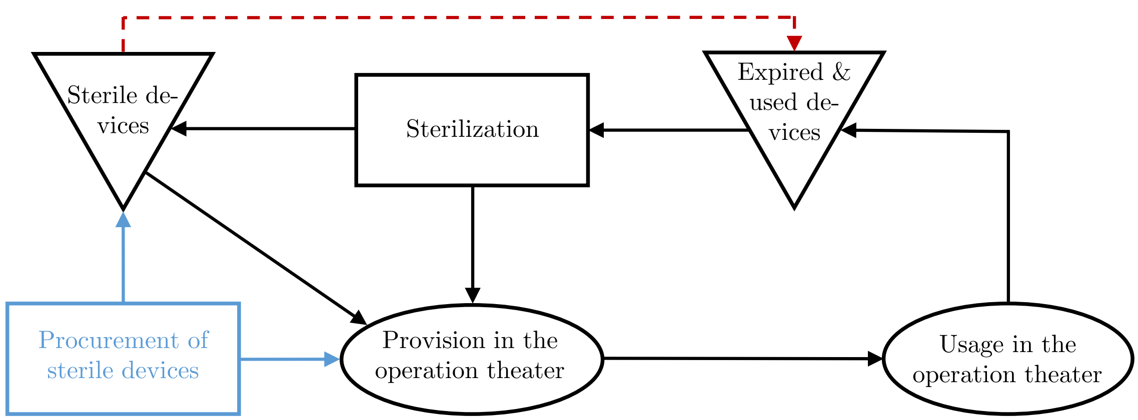

Figure 1: Product flow within the central sterile service in hospitals

In the following, the underlying assumptions for procurement, sterilization, storage, and demand fulfillment are introduced briefly:

- A procurement process causes fixed ordering costs and variable material costs.

- In each period, the ordering capacity is limited. In addition, a minimum order size must be met.
- Several sterilization types are available to reprocess medical devices. Medical devices can be sterilized by different types.
- For each sterilization process, fixed costs occur. Furthermore, variable unit costs are also taken into account.
- The capacity of the sterilization resource is restricted in each period.
- The storage time of sterile medical devices is restricted by legal requirements. Therefore, the storage time is recorded and the maximal storage time must be taken into consideration.
- The distinction between sterile and used medical devices requires a two-stage storage system.
- The storage capacity is restricted for both sterile and used/expired devices.
- An external demand d_{kt} for medical device k in period t has to be satisfied completely while backlogging is not allowed.
- A device-specific safety stock must be taken into account due to legal requirements.
- A portion $(1-\beta_k)$ of medical device k returns defectively and must be disposed of.
- Medical devices return with a time-lag of two periods. Thus, the returns r_{kt} of medical device k in period t are equal to $r_{kt} = \beta_k \cdot d_{k,t-2}$.

The objective of the OORPP is to determine ordering and reprocessing quantities that minimize the total sum of procurement and sterilization costs.

3 Outline of the Solution Approach

Following [3], the problem is solved in a heuristic fashion in two steps. First, we reformulate the underlying problem as a set partitioning problem and a column generation approach is applied to tighten the lower bound, see also [1] and [2]. In the next step, the obtained lower bound is transferred into a feasible solution by a truncated branch-and-bound approach using CPLEX.

4 First Numerical Results

To test the quality of the proposed solution approach, four problem classes (PC) were generated by varying the number of devices, periods and sterilization types. In total, 768 test instances were tested. Table 1 gives an overview of the numerical results of our solution approach.

	TCPU^{CG}	KFixed	TLim$^{B\&B}$	IntGAP
PC 1 ($K = 10, T = 12$)	$45s$	$35,1\%$	$100s$	$0,64\%$
PC 2 ($K = 20, T = 24$)	$237s$	$33,0\%$	$200s$	$2,13\%$
PC 3 ($K = 40, T = 24$)	$455s$	$55,3\%$	$400s$	$2,99\%$
PC 4 ($K = 80, T = 24$)	$811s$	$74,4\%$	$800s$	$3,83\%$

Table 1: Numerical results of the upper bounds

First, the average solution time in seconds to generate a feasible lower bound for the OORPP is reported in column "TCPU^{CG}". The entries in column "KFixed" show the portion of medical devices with an integer solution in the lower bound. The given time limit for the branch-and-bound approach is shown in "TLim$^{B\&B}$". Finally, the average integrality gap compared to the lower bound obtained by column generation is reported in "IntGAP".

The average solution time required for generating lower bounds is quite low. As expected, the computational effort increases with respect to the rising number of medical devices. The portion of medical devices with an integer solution is quite high with $35,1\%$ for PC 1. This portion rises to more than 70% for PC 4. We observe that the average integrality gap does not exceed 4% for all PC. Consequently, the results of this numerical study underline the high solution quality of the proposed solution approach.

References

[1] Cattrysse, Dirk; Maes, Johan and van Wassenhove, Luk N., Set partitioning and column generation heuristics for capacitated dynamic lotsizing, European Journal of Operational Research, 46, 38-47 (1990)

[2] Degraeve, Zeger and Jans, Raf, A New Dantzig-Wolfe Reformulation and Branch-and-Price Algorithm for the Capacitated Lot-Sizing Problem with Setup Times, Operations Research, 55, 909-920 (2005)

[3] Sahling, Florian, A Column-Generation Approach for a Short-Term Production Planning Problem in Closed-Loop Supply Chains, BuR - Business Research, 6, 55-75 (2013)

Multi-Item Bi-Level Capacitated Lot-Sizing with Remanufacturing of Reusable By-Products

Mehdi Rowshannahad, Nabil Absi, Stéphane Dauzère-Pérès
Department of Manufacturing Sciences and Logistics - CMP
CNRS UMR 6158 LIMOS - École des Mines de Saint-Étienne
rowshannahad@emse.fr, absi@emse.fr, dauzere-peres@emse.fr

Bernard Cassini
Soitec
Bernard.cassini@soitec.com

1 Introduction

Silicon wafers are extensively used in semiconductor manufacturing to produce microelectronic components such as chips and integrated circuits. However, some devices require higher performance which cannot be delivered by traditional silicon-only wafers. Components built on Silicon-On-Insulator (SOI) wafers offer much more performance while consuming less energy compared to components on silicon-only wafers. SOI wafers may be produced using different technologies. The production system studied in this paper concerns a type of *wafer bonding* technology called the *Smart-Cut™ Technology.* Using this technology, a thin layer of crystalline material is transferred from a donor substrate to another substrate using bonding and layer splitting processes. The used donor substrate can be processed later to be reused again as the donor substrate to produce another SOI wafer. Here, the *used donor wafer* is considered as a "by-product", i.e. it has been generated during production.

The studied manufacturing system in this paper has similarities to a *Closed-Loop Supply Chain* (*CLSC*). In order to classify our problem, we define a *Closed-Loop Manufacturing System* as a system in which by-products, co-products and/or components can be returned to the manufacturing system after a process called *product recovery.* Product recovery aims at restoring products while eliminating waste to a large degree. Here, we consider a specific type of *by-product recovery.* The industrial jargon of the company Soitec with for this type of recovery is *refreshing.*

The abstract is organized as follows. The main aspects of the Smart-Cut™ Technology and the studied manufacturing system are briefly described in Section 2, together with a short literature review. The main characteristics of the proposed mathematical model for planning production are summarized in Section 3. This model has been validated on test instances which have been generated from industrial data. Numerical experiments and managerial insights will be discussed in the workshop, together with some conclusions and perspectives of this study.

2 Problem Descrition and Literature Review

We consider the supply chain of a Silicon-on-Insulator (SOI) Wafer production unit using the Smart-Cut™ Technology. In SOI Wafers, a thin layer of silicon is laid on a silicon Wafer which serves only as a physical support (or handle). These two silicon layers are separated by an insulator: *The oxide*. Once Wafer **A** is oxidized and implanted, it is ready to be bonded with Wafer **B**. After the Wafers are bonded, they are split to form the SOI Wafer. Wafer **A** is the “donor” Wafer in the sense that a thin silicon layer of this substrate is deposited on Wafer **B**. In the industrial jargon, Wafer **A** is called “Top” while Wafer **B** is called “Base”. As only a thin layer of the Top Wafer is deposited on the Base Wafer, it is possible to reuse the Top Wafer several times to produce other SOI Wafers. This is one of the main advantages of the Smart-Cut™ Technology which makes the process cost competitive.

A “used Top Wafer”, called “Negative Wafer”, must be reworked before returning to the SOI fabrication process. This remanufacturing process is called the *refresh process* or shortly *refresh*. In the industrial terminology, a new Top Wafer used for the SOI fabrication is called a “Fresh Wafer” and is purchased from silicon Wafer suppliers. A Wafer may be refreshed only a maximum number of times (called the “maximum refresh level”). It is economically interesting to refresh a Top Wafer as many times as possible.

Yield and quality constraints or specific customer specifications complicate the SOI-Refresh planning. The refresh process, of limited capacity, can be done internally or externally. Internal refresh may also be performed in a different site than the one where the SOI Wafers are produced. Therefore, the increased cycle time (shipping and extra packaging) and possible deterioration must be considered. The studied problem is related to different domains of lot-sizing. We consider the production planning of a multi-item and multi-level manufacturing system. In our research, the raw material once used for production is considered as “by-product”. This by-product cannot fulfill any demand and must be reworked before coming back to the manufacturing cycle. The process of restoring the generated by-products makes them reusable again as raw materials. Therefore, this process can be considered as a “remanufacturing process”. In the literature, remanufacturing concerns the returned products from the customers whereas, in our case, the customer is not involved.

A multi-item uncapacitated lot-sizing problem in which co-products are produced at each production run is treated in [1]. In this paper, it is considered that the co-products have their own demand and cannot fulfill the demand of the main product. Note that in our case, by-products do not satisfy directly any demand. Besides, they are mostly used as newly purchased raw materials in the final product fabrication.

Remanufacturing in reverse logistics is considered in [3]. A closed-loop supply chain with setup costs, product returns and remanufacturing is considered in [5]. The study is inspired from the paper manufacturing industry in which both virgin and deinked pulps are used to make papers. Several studies use the term remanufacturing to denote the recycling of products or generated by-products ([2], [4]).

Our problem is different from the mentioned problems. It is bi-level, the by-products are generated when manufacturing the final products. Therefore, the generated quantities of by-products are deterministic. Moreover, there are multiple remanufacturing cycles, which depend on the raw material or the final product. Therefore, the decisions of purchasing, manufacturing and remanufacturing are connected.

3 Summarizing the Mathematical Model

To solve this complex industrial problem, we proposed a closed-loop lot-sizing model on a discrete finite time horizon. Because it is quite heavy, this model cannot be presented in this abstract. Instead, its main characteristics are discussed. This model uses big time bucket periods as multiple items can be produced in the same time period. The objective of our model is to decide when and how much to produce final products (SOI), when and how much to purchase raw materials, when and how much to refresh used Top Wafers in order to satisfy demand.

General parameters include the number of final products, Top (Fresh and Refresh) Wafers, Base Wafers, periods in the horizon and production or refresh sites, the demands of final products in each period, the maximum refresh level of each Top Wafer, the lead time of the refresh process, and the yield of the refresh process for each Top Wafer at each site. Parameters related to the bill-of-materials linking the different wafer types are required. Resource parameters include the refresh resource capacity at each site in each period, the process time of refreshing one unit of each Top Wafer at each site, the production capacity in each period and the process time of one unit of each product. Parameters related to initial inventory levels are also necessary for Base Wafers, Top Wafers, Negatives of Top Wafers (used Fresh or Refresh Wafers) and of final products. Finally cost parameters include inventory costs (for all wafer types), purchase costs (for Top Wafers and Base Wafers), refresh costs (for Top Wafers and Negatives), production costs (of final products) and setup costs (for purchasing Fresh Wafers and Base Wafers, for refreshing and for producing).

General variables include the production quantities of final products in each period, the ordered quantities of Base Wafers and Fresh Wafers in each period, the refreshed quantities obtained from Negatives at each site and in each period and the used quantities of Base Wafers and Top Wafers in each period. Inventory variables model the inventory levels of the final products, Base Wafers, Top Wafer and Negatives. Finally, setup variables are used to model if production occurs for each final product in each period, if procurement occurs for each Base Wafer and each Fresh Wafer in each period, and if the refresh process is performed in each site and in each period.

The objective function minimizes the total cost which is the sum of multiple types of purchase costs, production costs and setup costs. Flow conservation constraints are necessary for finished goods, based on the inventories of Base Wafers and Top (either Fresh or Refresh) Wafers, Base Wafers, Negative Wafers, Fresh Wafers and Refresh Wafers. These last flow conversation constraints include a yield factor since all of the Negative Wafers are not transformed into Refresh Wafers. Capacity constraints are used to respectively limit the production line and refresh line capacities in each period. Setup constraints are included to respectively model the production setup cost, raw material (Base Wafers and Fresh Wafers) procurement cost and refresh setup cost.

The primary novel feature of the problem is the refresh process. The remanufactured by-products (Refresh Wafers) and the newly purchased raw materials (Fresh Wafers) play the same role, i.e. they satisfy the raw material requirements for manufacturing the final product. Note that what makes the refresh process economically interesting is the lower cost of Refresh Wafers in comparison to purchasing Fresh Wafers.

Acknowledgments

This study has been done within the framework of a joint collaboration between SOITEC (Bernin, France) and the EMSE (Gardanne, France). The authors would like to thank the ANRT which has partially financed this study.

References

[1] Agrali, S. A Dynamic Uncapacitated Lot-Sizing Problem with Co-Production. *Optimization Letters 6*, 6 (Feb. 2011), 1051–1061.

[2] Ferrer, G., and Whybark, D. C. Material Planning for a Remanufacturing Facility. *Production and Operations Management 10*, 2 (2001), 112–124.

[3] Helmrich, M. J. R., Jans, R., van den Heuvel, W., and Wagelmans, A. P. Economic Lot-Sizing with Remanufacturing: Complexity and Efficient Formulations. *IIE Transactions 46*, 1 (2014), 67–86.

[4] Spengler, T., Puechert, H., Penkuhn, T., and Rentz, O. Environmental Integrated Production and Recycling Management. *European Journal of Operational Research 97*, 2 (Mar. 1997), 308–326.

[5] Zhang, Z.-H., Jiang, H., and Pan, X. A Lagrangian Relaxation Based Approach for the Capacitated Lot Sizing Problem in Closed-Loop Supply Chain. *International Journal of Production Economics 140*, 1 (Nov. 2012), 249–255.

Large-Scale Supply Chain Production Planning at JDA Software

Eric Prescott-Gagnon
JDA Innovations Labs
eric.prescottgagnon@jda.com

Vincent Raymond
JDA Innovations Labs
vincent.raymond@jda.com

Marc Brisson
JDA Innovations Labs
marc.brisson@jda.com

Abstract

JDA provides a production planning software for large companies that have global supply chains. At the heart of it is an optimization algorithm that must handle a wide variety of different features and constraints while optimizing on a large number of different competing objectives. The JDA Innovation Labs in Montreal is investigating new approaches such as column generation based on an hypergraph modelisation of the supply chain network to solve larger and larger real life instances.

1 Introduction

JDA offers software to help supply chain planners to plan the production at a high level. As supply chains grow more complex and need to handle more and more products, there is a need for algorithms that can scale and handle a large number of different features. Algorithms that can perform well on very different business needs coming from a wide variety of domains such as automotive, electronics, retail, food industry, etc. Production planning must be done on multiple products at the same time with production networks than span on multiple levels. Production is also constrained by resources that can be shared between operations. Safety-stock levels must also be sastisfied at each inventory point to ensure certain service levels. Many different objectives are also competing and must be handled simultaneously. The list of features goes on.

At JDA Innovation Labs in Montreal, we are investigating new ways to solve such large real life problems. We are investigating using a column generation approach to solve an hypergraph model of the problem.

2 Problem

The goal of the problem is to determine the flow of products or items inside the supply chain. Those products can be consumed at assembly plants to produce new intermediate or finished products. They can also be transported from one location (warehouse, supplier, assembly plant, etc.) to the other or kept in inventory at a given location. Given multiple suppliers for raw materials, different assembly plants with production rates, inventory policies, capacities and so on, the problem is to determine the optimal production to meet the demands of finished products.

A given inventory for a given product at a specific location can be refered to as a buffer. Buffers can have multiple safety stock levels defined. Operations consume materials from one or more specific buffers and produce new materials into new buffers. Example of operations can be to assemble two materials at an assembly plant and produce a new item. The operation would consume from the two buffers corresponding to the initial products and produce into the buffer corresponding to the product that was created. Transporting a product between two locations can also be represented as an operation consuming from a buffer and producing into another buffer of the same product at a different location. Operations can also have lot-sizing constraints such that products can only be produced in batches.

Operations consume resources that have a limited capacity at each time period. Resources can be machines or employees performing the operations. They can be shared between operations and some operations can have a choice of alternative resources if their primary resource is not available. Some resources are flexible, meaning that their capacity can be exceeded with a cost. For instance, if an employee were to work overtime can be modeled with a flexible resource.

An input to the problem is a hierarchical list of objectives. The first objective on the list being infinitely more important than the next ones and so on. The list is different for each instance and depends on the business needs of the supply chain planner. Here is a list of some of the objectives than can be configured:

- Satisfy demands as much as possible
- Minimize backlogged demands
- Satisfy safety stock levels
- Favor just-in time operations
- Minimize use of alternative resources
- Minimize use of exceeded resource capacities
- Etc.

3 Approaches

The supply chain network can be modeled as an hypergraph. Each node in the hypergraph is a buffer at a specific time period. The graph is thus a copy of the supply chain network for each time period in the planning horizon. For each buffer, there are arcs that link between each node of a time period to the node corresponding to the next time period. These arcs represent inventory that is carried over from one period to the next. Operations can be represented as hyperarcs that may have multiple in or out nodes depending on from how many buffers they are consuming and into many buffer they are producing. We get the following formulation:

$$min \sum_{b \in B} \sum_{t \in T} c_{bt} s_{bt} + \sum_{o \in O} \sum_{t \in T} c_{ot} x_{ot} + \sum_{d \in D} \sum_{t \in T} c_{dt} q_{dt} \tag{1}$$

$$s_{b,t-1} - s_{bt} + \sum_{o \in O} a^{+}_{ob} x_{o,t-l^{-}_{ob}} - \sum_{o \in O} a^{-}_{ob} x_{o,t+l^{+}_{ob}} - \sum_{d \in D} a_{db} q_{dt} = 0, \quad \forall b \in B, t \in T \tag{2}$$

$$\sum_{o \in O} a_{or} x_{ot} \leq C_{rt}, \quad \forall r \in R, t \in T \tag{3}$$

$$\sum_{t \in T} q_{dt} \leq Q_d, \quad \forall b \in B, t \in T \tag{4}$$

B,O,D and T are the sets of buffers, operations, demands and time periods respectively. We define the variables s_{bt} are the amount of inventory remaining in buffer b at the end of time period t, x_{ot} as the amount of operation o performed at time period t, and q_{dt} as the amount of demand d satisfied at time period t. We define the costs c_{bt}, c_{ot} and c_{dt} as carryover costs, operation costs and demand costs, respectively. The value of the costs will depend on what objective is being solved. Other coefficients include a^{+}_{ob} (a^{-}_{ob}) as the amount of inventory of buffer b produced (consumed) by one unit of operation o, l^{+}_{ob} (l^{-}_{ob}) is the production (consumption) leadtime of operation o with respect to buffer b, a_{db} is 1 if demand d can be satisfied by buffer b, 0 otherwise, and a_{or} is the amount of resource r consumed by one unit of operation o. Finally, C_{rt} and Q_d are resources capacities and demand quantities respectively.

Equation (1) is the objective function, (2) are flow conservation constraints at each node of the hypergraph, (3) are resource capacity constraints and (4) are demand satisfaction constraints. Note that one of the objectives is to maximise the amount of demand satisfied, so equation (4) prevent oversatisfaction of demands.

The JDA Innovation Labs in Montreal are investigating new ways to solve the problem using decomposition such as column generation. A column generation reformulation of the problem based on hyperpaths was investigated. In that formulation, an hyperpath, is a path containing arcs and hyperarcs that takes one unit of inventory out of the network by either satisfying one unit of demand or by carrying-over one unit of inventory out of the planning horizon. The master problem becomes a problem of assembling a set of hyperpaths maximising the objective and respecting the resource and demand constraints. The subproblem generates new hyperpaths in the network and can be solved by dynamic programming.

Two-Level Lot-Sizing with Raw-Material Perishability and Deterioration: Formulations and Analysis

Andrés Acevedo-Ojeda
Concordia University and CIRRELT, Montreal, Canada, H3G 1M8
a_aceved@encs.concordia.ca

Ivan Contreras
Concordia University and CIRRELT, Montreal, Canada, H3G 1M8
icontrer@encs.concordia.ca

MingYuan Chen
Concordia University, Montreal, Canada, H3G 1M8
mychen@encs.concordia.ca

Abstract

In many industries, it is common to face significant rates of product deterioration, referring not only to physical exhaustion or loss of functionality, but also obsolescence. We study how raw-material perishability and deterioration enforces specific constraints on a set of production planning decisions, especially in the case of multi-level product structures. We study three variants of the two-level lot-sizing problem with raw-material *shelf-life* and deterioration. Finally, an extended MIP formulation is proposed and sensitivity analysis is carried out to better understand the benefits and relevance of our model in comparison with traditional lot-sizing models.

1 Introduction

A common assumption in most of the production planning literature is that the products involved in the production process have unlimited *lifespans*, meaning they can be stored and used indefinitely. However, in practice, most items deteriorate over time, referring not only to physical exhaustion or loss of functionality, but also obsolescence. Often, the rate of deterioration is low or can be ignored and there is little need for considering it in the planning process. Nonetheless, in many types of industries, it is common to deal with items that are subject to significant rates of deterioration. These items are referred to as *perishable products*. The concept of *perishability* basically relates to items that cannot be stored infinitely without deterioration or devaluation

[1]. Clear cases of this type of products can be found in the food or pharmaceutical industries [2, 3]. For instance, in the yogurt industry [4], perishability is found in different levels of the production process: from the highly perishable raw-material (milk) that enters the dairy factories, to the finished-products, which are all stamped with a best-before-date fixing its *shelf-life*.

As mentioned in [5], perishability and deterioration enforce specific constraints on a set of crucial production planning decisions, specially in the case of multi-level production structures, where two or more items are produced, and at least one item is required as an input (raw-material, component, part) of another. These intermediate products, either acquired from a supplier or processed internally, can often be inventoried, allowing one to produce and consume them at different moments and rates in time [6]. When dealing with perishability, most of the data associated with inventories has to be extended in order to track the age and usability status of items with specific time-stamps. Besides the amount of inventory kept in stock, we also need to know when the material has been acquired and to what level it has deteriorated, as well as the impact that such deterioration may have in the production process.

In this study, we propose to incorporate raw-material perishability considerations into classical production planning problems. We evaluate how to integrate restrictions on shelf-life and deterioration of intermediate items. We formulate such problems as mixed-integer programs (MIP), and model the impact that perishability may have in the production process regarding aspects such as: manufacturing, inventory, and disposal costs; and the quality of the finished-products. We study three variants of a two-level lot-sizing problem involving different types of perishability, and perform sensitivity analysis to better understand the benefits and relevance of our models in comparison with traditional lot-sizing.

2 Lot-sizing with perishable raw material

Consider a production system in which one *finished-item* is to be produced and another item (*raw-material*), required as an input of the first, is to be procured from a supplier. This constitutes the simplest version of a two-level production structure. In general terms, the two-level lot-sizing (*2L/LS*) problem is to find the production, procurement, and inventory plan for the two items over a discretized planning horizon (divided into n time periods, where $T = \{1, ..., n\}$), meeting finished-item demand in every period, minimizing the corresponding costs. As mentioned above, the core aspect of the problems under study is the perishable condition of the raw-material. In this regard, we study three different variants of the *2L/LS* problem: (a) raw-material subject to a fixed shelf-life (*FSL*), with constant functionality; (b) raw-material subject to a functionality deterioration rate (*FDR*) and; (c) raw-material subject to functionality and volume deterioration rate (*FVDR*). Each of these cases is described in the following sections.

2.1 Fixed Shelf-Life

The first variant of our problem is what we call the two-level lot-sizing problem with fixed raw-material shelf-life (*2L/LS/FSL*). Raw-material orders are acquired in batches from an external supplier under the immediate receipt assumption (no ordering lead-time). Once the raw-material is received, it is used to satisfy finished-item production requirements or can be inventoried. However, due to perishability, on-hand raw-material can only be kept in stock and used for a fixed period of time (shelf-life). If the material reaches the end of its shelf-life and expires, it will have to be disposed. This causes additional costs that can vary depending on when the disposal is made. Raw-material functionality is considered to remain constant during the entire shelf-life period.

Potential applications of this problem can be found in the production of plastic films. A plastic film is a thin continuous polymeric material used to separate areas or volumes, to hold items, to act as barriers, or as printable surfaces. Plastic films are used in: food and nonfood plastic packaging films, plastic bags, labels, and photographic films, are some of the applications, among other applications. Depending on the properties and characteristics of the desired application, plastic films can be made from a variety of plastic resins and monomers, which are highly reactive and undergo uncontrolled polymerization (reason why we consider them perishable). As raw-material for plastic films, these resins and monomers are considered fully functional during their shelf-life and, once they are used in production, they become stable. Furthermore, the finished-items (the plastic films), even though they may present sings of deterioration during the storage and handling period, they have shelf-lives long enough not to be considered perishable.

2.2 Functionality deterioration

The second variant of our problem is the two-level lot-sizing problem with raw-material functionality deterioration rate (*2L/LS/FDR*). Here, in addition to having on-hand raw-material subject to a fixed shelf-life, its functionality, utility or efficiency level, decreases progressively as storage time passes. In this specific case, although the shelf-life remains fixed, it is determined by the number of periods it takes for the material to completely lose its functionality. Progressive decrease in material functionality has a direct impact on production. This impact may be, for instance, on the quality of the finished-items produced with deteriorated material, or an increase in the production cost in order to reach the same desired quality or production yield, using material that is not fully functional. Either way, we represent such impact by an increase in the production cost. Potential applications of this problem can be found in the composite manufacturing and related industries, when producing polyimide reinforced fiber composites and other products.

2.3 Functionality and volume deterioration

In this specific case, we propose the two-level lot-sizing problem with raw-material functionality and volume deterioration rates ($2L/LS/FVDR$). Here, the perishability nature of the raw-material is biphasic, referring not only to a functionality loss but, in addition, to a progressive volume loss. In this sense, we have a *volume deterioration rate*, representing the fraction of the material that is lost in each period of time. Applications of this problem can be found in the so-called canning industry, in production processes such as: canning fruits, vegetables, juices, fish and seafood, meats; processing ketchup and other tomato sauces; and producing natural and imitation preserves, jams and jellies. The primary objective of food processing is the preservation of highly perishable goods in a stable form that can be stored and shipped to distant markets. Canning provides a shelf-life typically ranging from one to five years, although under specific circumstances it can be much longer. However, it is normal to face considerable levels of raw-material loss throughout the multiple steps of the production process, which includes preliminary preparation, blanching, and filling.

References

[1] Billaut, Jean-Charles, New scheduling problems with perishable raw materials constraints, IEEE 16th Conference on Emerging Technologies and Factory Automation (ETFA), 1–7 (2011)

[2] Farahani, P., Grunow, M., and Gunther, H.-O., Integrated production and distribution planning for perishable food products, Flexible Services and Manufacturing Journal, 24, 28–51 (2012)

[3] Vila-Parrish, Ana R., Ivy, Julie Simmons, and King, Russell E., A simulation-based approach for inventory modeling of perishable pharmaceuticals, IEEE 2008 Winter Simulation Conference, 1532–1538 (2008)

[4] Entrup, Lütke M., Günther, H.-O., Van Beek, P., Grunow, M., and Seiler, T., Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production, International Journal of Production Research, 43, 5071–5100 (2005)

[5] Amorim, P., Meyr, H., Almeder, C. and Almada-Lobo, B., Managing perishability in production-distribution planning: a discussion and review, Flexible Services and Manufacturing Journal, 25, 389–413 (2011)

Lower and Upper Bounds for The Integrated Lot-Sizing and Cutting Stock Problem

Gislaine Mara Melega
Unesp - IBILCE, Universidade Estadual Paulista, Brazil.
gislainemelega@gmail.com

Silvio Alexandre de Araujo
Unesp - IBILCE, Universidade Estadual Paulista, Brazil.
saraujo@ibilce.unesp.br

Raf Jans
HEC Montreal and GERAD, Canada H3T 2A7 QC, Canada.
raf.jans@hec.ca

Abstract

In this work, we approach the lot-sizing and the cutting stock problems in an integrated way. We propose different mathematical models for the integrated problem considering different models for the lot-sizing and the cutting stock problem, in order to evaluate and indicate the impact of these changes on the models' performance. An extensive computational study was done using randomly generated data and as a solutions strategy we used a commercial optimization package and the application of a column generation technique.

1 Introduction

In industrial sectors such as furniture, paper and aluminum, the lot-sizing and cutting stock problems are found in consecutive phases. For this reason, the relevance of these issues in the industrial sectors and the benefits in dealing with problems in an integrated way makes this an appropriate research topic today. In the literature, integrated problems are studied for instance in [3], [6], [4]. Motivated by this, the present paper proposes mathematical models and solution methods for solving the integrated lot-sizing and cutting stock problem.

The lot-sizing problem considers the tradeoff between the setup and inventory costs to determine at minimal cost the size of production lots to meet the demand of each final product. In this work, the lot-sizing part is modeled using the mathematical model proposed by [7], denoted here by *CL*. We also considered the variable redefinition strategy ([1]) which reformulates the lot-sizing problem as a shortest path problem. This reformulation is called *SP*.

The cutting stock problem involves the cutting of large objects into smaller items, so as to minimize the total loss of material. We consider three mathematical models from the literature to model the cutting stock problem. The first one is the model developed by [5], here denoted by *KT* and it determines the best way to cut objects to meet the items demand, minimizing the number of objects used. The second model dealt with, and perhaps the best known among the academic community, is the one proposed by [2] denoted here by *GG*. The third model was proposed by [8], here denoted *VC*. The author proposes an alternative mathematical model for the one-dimensional cutting stock problem based on an arc flow problem. Furthermore, the cutting stock models above were originally proposed for a single object type and one time period, in this way they are extended to consider several types of objects in stock (*MO* from multi-objects) and multi-periods.

2 Mathematical Models

We consider all possible combinations of lot-sizing formulation (*CL*, *SP*) and cutting stock formulations (*KT*, *GG* and *GG*). We also analyze the cutting stock problem with several types of objects (*MO*), which is combined with the uncapacitated lot-sizing problem ([9]). Again we consider the combinations of various formulations.

In order to illustrate the idea of the integrated models, we present the *CLGG* integrated model, that consists in the integration of the *CL* model and the *GG* model. The capacity constraint is considered in terms of end items, which means that the capacity constraints are related to the final production process to transform the cut items into the final items.

For this consider $t = 1, ..., T$ as the times periods, $i = 1, ..., I$ as the number of items and $j = 1, \dots, J$ as the set of cutting patterns. We define sc_{it} as the setup cost, hc_{it} as the unit holding cost, co as the cost of the objects, st_{it} as the setup time, vt_{it} as the production time, Cap_t as the capacity available and a_{ij} as the number of objects i cut in cutting pattern j.

The decision variables are as follows: X_{it} is the production quantity of item i in period t, S_{it} is the inventory for item i at the end of period t, Y_{it} is a binary variable indicating the production or not of item i in period t and x_{jt} indicates the number of objects cut according to cutting pattern j in period t;

$$min \sum_{t=1}^{T}\sum_{i=1}^{I}(sc_{it}Y_{it} + hc_{it}S_{it}) + \sum_{t=1}^{T}\sum_{j=1}^{J} cox_{jt} \tag{1}$$

Subject to :

$$X_{it} + S_{it-1} - S_{it} = d_{it} \qquad \forall i,\ \forall t \tag{2}$$

$$X_{it} \leq sd_{itT}Y_{it} \qquad \forall i,\ \forall t \tag{3}$$

$$\sum_{i=1}^{I}(st_{it}Y_{it} + vt_{it}X_{it}) \leq Cap_t \qquad \forall t \tag{4}$$

$$\sum_{j=1}^{J} a_{ij}x_{jt} = X_{it} \qquad \forall i,\ \forall t \tag{5}$$

$$Y_{it} \in \{0, 1\} \qquad \forall i,\ \forall t \tag{6}$$

$$x_{jt} \in \mathbb{Z}_+ \qquad \forall j,\ \forall t \tag{7}$$

$$X_{it},\ S_{it} \in \mathbb{R}_+ \qquad \forall i,\ \forall t \tag{8}$$

The objective function (1) minimizes the sum of setup costs, inventory costs and the cost related to the number of objects used in the cutting process. The sets of constraints (2), (3) and (4) refer to the lot-sizing problem. Constraints (5) link the cutting variable with the production variable, that is, it is necessary to cut a sufficient amount of items to meet the amount that is desired to produce. Finally the constraints (6), (7) and (8) are non-negativity and integrality constraints on the variables.

3 Computational Study

This section describes the solution methods used to solve the proposed models, as well as a summary of the computational results obtained, which evaluated and compared the performance of the proposed models. The models are written in the AMPL syntax and the CPLEX 12.5 was used as solver. All the computational tests were conducted on a 2.93GHz Intel Core i7 processor with 8GB of RAM memory.

- **Solution Strategy:** Two heuristic strategies are proposed for solving the formulations described in section 2. The first heuristic consists of solving the models integrated with *KT* and *VC* by establishing some stop criteria for the optimization package. The solver is stopped at the end of 600 seconds or when the gap between the upper bound and lower bound is less than 0.1%. The second heuristic uses the column generation technique for the models integrated with *VC* and *GG* to obtain a lower bound. The column generation is stopped when the minimum reduced cost is non-negative or when the computation time limit is reached. The optimal columns are fixed and the integer model is solved in order to obtain a feasible solution. For the integer problem we set the processing time to 600 seconds and the optimality gap equal to 0.1%.

- **Analysis of the Results:** The results showed the clear difficulty of the models to obtain a feasible solution when considering a capacity constraint in the lot-sizing problem, and this impact is even greater when the capacity constraint is tight. Instances that have an item length considerably smaller compared to the object length are much more difficult to solve. For the models that consider various object types in stock, the major fact that influences the performance is the number of items. The models integrated with the classical lot-sizing problem and the cutting stock problem model *VC* based on [8] and column generation obtained the largest number of feasible solutions compared to other mathematical models in all analyzes. The models that integrate with the model *KT* proposed by [5] produced bad results in all analyzes. The use of the shortest path model significantly improved the lower bound and resulted in gaps smaller when compared to other models, for most classes.

References

[1] Eppen, G. D.; Martin, R. K., Solving multi-item capacitated lot-sizing problems using variable redefinition, Operations Research, 35, 832848 (1987).

[2] Gilmore, P. C.; Gomory, R. E., A linear programming approach to the cutting-stock problem, Operations Research, 9, 849 859 (1961).

[3] Gramani, M. C. N.; Frana, P. M., The combined cutting stock and lot-sizing problem in industrial processes, European Journal of Operational Research, 174, 509 521 (2006).

[4] Gramani, M.; Frana, P. M.; Arenales, M. N., A linear optimization approach to the combined production planning mode, Journal of the Franklin Institute, 348, 1523 1536 (2011).

[5] Kantorovich, L. V., Mathematical methods of organizing and planning production, Management Science, 6, 366 422 (1960).

[6] Poltroniere, S. C.; Poldi, K. C.; Toledo, F. M. B.; Arenales, M. N., A coupling cutting stock-lot sizing problem in the paper industry, Annals of Operations Research, 157, 91104 (2008).

[7] Trigeiro,W.W.; Thomas, J.; McClain, J. O., Capacitated lot sizing with setup times, Management Science, 35, 353366 (1989).

[8] Valerio de Carvalho, J., Exact solution of bin-packing problems using column generation and branch-and-bound, Annals of Operations Research, 86, 629 659 (1999).

[9] Wagner, H. M.; Whitin, T. M., Dynamic version of the economic lot size model, Management Science, 5, 8996 (1958).

Heuristic Methods for a Capacitated Lot-Sizing Problem with Stochastic Setup Times

Duygu Taş
HEC Montréal and CIRRELT
duygu.tas@cirrelt.ca

Michel Gendreau
École Polytechnique de Montréal and CIRRELT
michel.gendreau@cirrelt.ca

Ola Jabali
HEC Montréal and CIRRELT
ola.jabali@hec.ca

Raf Jans
HEC Montréal and CIRRELT
raf.jans@hec.ca

Abstract

We consider a capacitated lot-sizing problem with stochastic setup times. A mathematical model is described that considers both regular production costs and expected overtime costs. We propose an effective procedure to compute the expected overtime for a given production plan and develop several heuristics to generate efficient solutions. We provide extensive computational results and confirm that our solution approaches obtain very good solutions.

1 Introduction

In this study, we focus on a Capacitated Lot-Sizing Problem (CLSP) with stochastic setup times. In the CLSP, several different items can be produced in the same period where a capacitated single machine is used for all items. In other words, the total time spent for setup and production activities in each period is limited by the time capacity of the machine. This problem aims to obtain a production plan with the minimum total cost which satisfies the demands and respects the capacity of the machine. In our problem, we consider stochastic setup times at the planning level to provide efficient production plans to be employed at the operational level. To have a tractable problem, we assume that the random setup times follow suitable Gamma distributions. The capacity of the machine is considered as a soft constraint, which allows overtime usage by incurring some penalty costs. The overtime values can be thought of as recourse decisions, which are determined after observing the specific realization of setup times. Note that the production quantities and setups are decided at the beginning of the planning horizon (frozen schedule). This setting with no dynamic adjustment corresponds to the static uncertainty strategy given by Bookbinder and Tan [2]. The objective is to minimize the sum of regular production costs and expected overtime costs. In our model, regular costs result from three elements which are the total production quantity, the total number of setup and the total inventory. In addition, the expected overtime in each period is computed with respect to the production plan and capacity parameters, i.e., variable production times, setup times and capacity.

2 Heuristic Methods

Four heuristics are proposed in this study to solve the problem described above, where each method employs a specific formulation:

- The first heuristic is based on changing certain parameters (such as capacity and setup times) in a deterministic model with no overtime. More specifically, we consider both the standard formulation of the CLSP (see Trigeiro et al. [6]) and the transportation problem reformulation proposed by Krarup and Bilde [8]. The latter provides a better LP relaxation gap compared to the formulation in the original variables (see Pochet and Wolsey [7]). Note that when modified parameters are implemented, these models may result in infeasible solutions due to the production capacity limit.

- The second heuristic uses a model allowing planned overtime (see, e.g., Özdamar and Birbil [5], Özdamar and Barbarosoğlu [4], and Barbarosoğlu and Özdamar [1]) and further adjusts problem parameters such as the capacity and setup times. In these versions, deterministic overtime can be thought of as the amount added to the original capacity at the planning level. In other words, this method is always capable of ending up with a feasible solution by exceeding capacity in case it is needed.

- The third heuristic considers balancing the usage of capacity in each period. For this method, we propose a new formulation where the aim is to distribute the production of all items evenly over all periods and to generate a production plan resulting in a smaller expected overtime compared to that obtained by the standard formulations of the CLSP.

- The fourth heuristic employs a two-stage stochastic programming model with a limited number of scenarios. In the proposed formulation, the production and setup decisions are the first-stage decisions and the overtime corresponds to the recourse decision in the second-stage.

All solution approaches are based on solving the considered models (with either original or modified problem parameters) and then evaluating the obtained solution with respect to stochastic setup times.

3 Computational Results

Extensive computational results are obtained by experimenting with data sets given in Trigeiro et al. [6]. Each model described in Section 2 is coded in C++ and solved using IBM ILOG CPLEX 12.5 [3]. All experiments are performed on an Intel(R) Xeon(R) CPU X5675 with 12-Core 3.07 GHz and 96 GB of RAM (by using a single thread). We set a limit for the total computation time which is equal to 30 minutes.

In the first and second heuristics, problem instances are solved by employing the parameter settings with: (i) original values given for each problem element, (ii) a smaller capacity for each period and (iii) larger setup times. In (ii) and (iii), we aim to provide some buffer capacity to be used in the stochastic environment, which possibly leads to smaller expected overtime. Results show that changing problem parameters at the deterministic planning level yields production plans which perform better in the stochastic environment compared to those obtained by the models with original problem elements. Specifically, the deterministic model allowing planned overtime with increased setup times leads to more robust solutions (with an improvement in the total expected cost by 3.00% on average) with respect to those obtained by the classical CLSP. Compared to the solution of the standard formulation, (i) when we increase the variation in setup times, this heuristic leads to an improvement of 11.51% on average, and (ii) when we increase the cost of overtime, this heuristic leads to an improvement of 27.75% on average.

Results obtained by the third type of heuristic indicate that balancing the usage of capacity leads to smaller expected overtime and further smaller total expected costs compared to the solutions generated by the classical models. However, the overall performance of this method underperforms with respect to that of the first and second heuristics.

In the fourth heuristic, the number of scenarios takes values from 10 to 100. We observe that the final optimality gap and the required computation time increase as the number of scenarios increases. This approach provides the smallest total expected cost over all instances, where the provided improvement with respect to the total expected cost of the classical CLSP model is 3.09% on average. Compared to the solution of the standard formulation, (i) when we increase the variation in setup times, this heuristic leads to an improvement of 12.68% on average, and (ii) when we increase the cost of overtime, this heuristic leads to an improvement of 27.72% on average.

Detailed analyses show that the heuristic based on a two-stage stochastic programming (the fourth heuristic) provides very good solutions to be employed in stochastic settings, especially when we have high variability in setup times. We observe that the heuristic based on the model with planned overtime where setup times are modified with respect to the Gamma distribution (the second heuristic) also performs well in the stochastic environment, especially when exceeding capacity brings a high violation cost. Finally, overall analyses confirm that these two effective heuristics provide similar performances in terms of the solution quality and the computation time for the considered stochastic problem which is a close representation of real-life applications.

References

[1] G. Barbarosoğlu, L. Özdamar, Analysis of solution space-dependent performance of simulated annealing: the case of the multi-level capacitated lot sizing problem, Computers and Operations Research, 27, 895–903 (2000).

[2] J.H. Bookbinder, J.-Y. Tan, Strategies for the probabilistic lot-sizing problem with service-level constraints, Management Science, 34, 1096–1108 (1988).

[3] IBM, ILOG CPLEX Optimizer 12.5, `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer` (2015).

[4] L. Özdamar, G. Barbarosoğlu, Hybrid heuristics for the multi-stage capacitated lot sizing and loading problem, Journal of the Operational Research Society, 50, 810–825 (1999).

[5] L. Özdamar, S.I. Birbil, Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions, European Journal of Operational Research, 110, 525–547 (1998).

[6] W. Trigeiro, L. Thomas, J. McClain, Capacitated lot sizing with setup times, Management Science, 35 353–366 (1989).

[7] Y. Pochet, L.A. Wolsey, Production planning by mixed integer programming, Springer (2006).

[8] J. Krarup, O. Bilde, Plant location, set covering and economic lot sizes: an O(mn) algorithm for structured problems, in: L. Collatz, G. Meinardus, W. Wetterling (Eds.), Optimierung bei Graphentheoretischen und Ganzzahligen Probleme, Numerische Methoden bei Optimierungsverfahren, Band 3, Birkhauser Verlag, Basel, pp. 155–179 (1977).

Dynamic Safety Stock Planning in the General Lot-sizing and Scheduling Problem

Stefan Minner
TUM School of Management, Technische Universität München, Arcisstrasse 21, 80333 Munich, Germany

Dariush Tavaghof-Gigloo
TUM School of Management, Technische Universität München, Arcisstrasse 21, 80333 Munich, Germany
dariush.gigloo@tum.de

Abstract

We present a mixed integer linear programming (MILP) approach to integrate dynamic safety stock planning into the general lot-sizing and scheduling problem (GLSP) model formulation with continuous and non-equidistant micro-periods. We consider a base stock policy with different service measures. We implement state-of-the-art univariate and bivariate linearization techniques for the non-linear first-order loss function. In an extensive experimental study, we compare our approach to its deterministic counterpart in a rolling horizon environment.

1 Introduction

The variable nature of the replenishment periods in lot-sizing problems and the non-linearity and non-separability of the first-order loss function complicate integrating safety stock approaches directly into lot-sizing problems. Hence, lot-sizing and safety stock placement problems are generally treated separately (sequentially) in the inventory and operations management literature. There exist integrated approaches but they are rather complex or usually limited to single item uncapacitated lot-sizing problems, e.g., see [1], [11], [12], and [10]. To the best of our knowledge, the stochastic version of the general lot-sizing and scheduling problem (GLSP) has not been properly addressed in the literature yet. The deterministic GLSP has been studied by [3], [8], [7], and [13]
In this paper, we develop an equivalent deterministic model to integrate non-stationary safety stock planning into the general lot-sizing and scheduling problem (GLSP) with continuous and non-equidistant micro periods. We assume a service level approach by

taking the non-stockout probability and fill-rate service measures into account. We introduce a new MILP approach to determine cumulative mean demand and variance of demand for the time spans between consecutive production events on the continuous micro periods. The order-up-to-level function can arise either as a univariate or a bivariate non-linear function, depending on the service measure and demand distribution. Hence, we use an efficient univariate linearization technique as well as a bivariate linearization technique based on the triangulation method. We conduct an extensive numerical study based on the randomly generated data sets to show the applicability of our approach and its superiority over its deterministic counterpart with stationary safety stock settings in a rolling horizon environment. In the following sections, we briefly describe our approach and the numerical experiments.

2 Equivalent Deterministic Model

We assume a deterministic single-level general lot-sizing and scheduling problem (GLSP) with discrete and equidistant macro-periods and continuous non-equidistant micro periods. Multiple products are produced on a single capacitated machine with sequence dependent set-up times. We first develop a standard MILP model formulation of the problem (also its reformulation, e.g., see [13]) with the objective of minimizing the total inventory holding costs under certain constraints. We assume a base stock policy where demand is non-stationary and independent with a known probability distribution. We extend the model to an equivalent deterministic model of a stochastic version of the described GLSP by introducing service level constraints on the continuous micro periods. Under the base stock policy, the order-up-to-level depends on the cumulative mean demand between consecutive replenishment periods, the corresponding variance of demand, demand distribution, and type and size of the service level.

We add new MILP model formulations to the equivalent deterministic model formulation to determine the cumulative mean demand and the cumulative variance of demand between consecutive replenishment periods. We further distinguish between continuous inventory depletion, where final products can be used to fulfill demand during production time as they are just produced, and closed depletion, where final products become only available after the whole batch is produced. In case of continuous inventory depletion, cumulative mean demand and variance of demand are determined for the time spans from the end of every production period to the start of the next production period.

In case of a non-stockout probability service level with normally distributed demand, we face with a univariate non-linear term with the square root of the cumulative variance of demand. To linearize this square root term, we test several linearization techniques like the convex combination, the incremental model, and the multiple

choice model as given in [2] and choose the one which provides the best computational performance. The accuracy of the approximation method is improved by determining appropriate lengths of the approximation segments, e.g., see [6].
In case of considering fill-rate or non-stockout probability with other distributions than normal, we face a bivariate, non-linear, non-separable first-order loss function depending on both, cumulative mean demand and cumulative variance of demand. For the linearization of this function, we use a triangulation method to construct an approximation grid which is divided into several triangles. The value of the order-up-to-level is approximated at a certain point by using three weights indicating the distance of the point to the vertices of its corresponding triangle. The point represents the combination of the cumulative mean demand and the cumulative variance of demand, e.g., see [9].

3 Experimental Study

We test the numerical behaviour of the proposed equivalent deterministic model in comparison to the deterministic GLSP with stationary safety stocks. For the equivalent deterministic model, we assume several service sizes for non-stockout probability and fill-rate. For the deterministic counterpart, we use different settings of stationary safety stock levels (high, medium, and low). We generate random data sets mainly based on the numerical designs which have been presented in [7], [4], and [5]. Both approaches are tested under a rolling horizon environment for every data set. The average total cost, including the total inventory costs and the lost-sales costs, as well as the average actual service sizes are reported. We further investigate the computational performance of our approach and the proposed linearization techniques which are embedded in the equivalent deterministic model for both the standard formulation and reformulations. Finally, we elaborate the consequences of using different initial inventory levels or capacity levels on the actual service sizes under a rolling horizon environment.

References

[1] Bookbinder, J. H., & Tan, J. Y., Strategies for the probabilistic lot-sizing problem with service-level constraints, Management Science, 34(9), 1096-1108 (1988)

[2] Croxton, K. L., Gendron, B., & Magnanti, T. L., A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems, Management Science, 49(9), 1268-1273 (2003)

[3] Fleischmann, B. and Meyr, H., The general lotsizing and scheduling problem, Operations-Research-Spektrum, 19(1), 11-21 (1997)

[4] Haase, K. and Kimms, A., Lot sizing and scheduling with sequence-dependent set-up costs and times and efficient rescheduling opportunities, International Journal of Production Economics 66(2), 159-169 (2000)

[5] Helber, S., Sahling, F., & Schimmelpfeng, K., Dynamic capacitated lot sizing with random demand and dynamic safety stocks, OR spectrum, 35(1), 75-105 (2013)

[6] Imamoto, A. and Tang, B., Optimal piecewise linear approximation of convex functions, Proceedings of the world congress on engineering and computer science, (2008)

[7] Koclar, A., The general lot sizing and scheduling problem with sequence dependent changeovers, Doctoral dissertation, Middle East Technical University, (2005)

[8] Koclar, A. and Sueral, H., A note on The general lot sizing and scheduling problem, OR Spectrum, 27(1), 145-146 (2005).

[9] Rebennack, S. and Kallrath, J., Continuous Piecewise Linear -Approximations for MINLP Problems, II. Bivariate and Multivariate Functions, 2012-13, (2012)

[10] Rossi, R., Kilic, O. A., & Tarim, S. A., Piecewise linear approximations for the static dynamic uncertainty strategy in stochastic lot-sizing, Omega, 50, 126-140, (2015)

[11] Tarim, S. A., & Kingsman, B. G., The stochastic dynamic production/inventory lot-sizing problem with service-level constraints, International Journal of Production Economics, 88(1), 105-119 (2004)

[12] Tempelmeier, H., & Herpers, S., Dynamic uncapacitated lot sizing with random demand under a fill-rate constraint, European Journal of Operational Research, 212(3), 497-507, (2011)

[13] Transchel, S., Minner, S., Kallrath, J., Löhndorf, N., & Eberhard, U., A hybrid general lot-sizing and scheduling formulation for a production process with a two-stage product structure, International Journal of Production Research, 49(9), 2463-2480, (2011)

A Bi-Objective Genetic Algorithm for Economic Lot Scheduling Problem Under Uncertainty

Canan Capa, Ali Akgunduz
Concordia University, Canada
c_capa@encs.concordia.ca, akgunduz@encs.concordia.ca

Abstract

Economic lot sizing problem (ELSP) is one of the most studied problems in production planning and it deals with the scheduling of a set of products in a single machine to minimize the long run average holding and setup cost under the assumptions of known constant demand and production rates. In this study, since the traditional deterministic ELSP models neglect the uncertainty of the demand rates and setup times and so may perform poorly under certain realizations of the random demand rates and setup times we consider the ELSP by exploiting the assumption of deterministic demand rates and setup times and formulate it as a bi-objective optimization problem that aims at providing solution robust production schedules. To solve the problem, we use a hybrid approach, combining a genetic algorithm with linear programming. The aim of this approach is, through K number of possible schedule realizations, to generate a set of non-dominated solution robust production schedules, i.e., production schedules with least deviation from the realized schedules in terms of makespan and total cost.

1 Introduction

Economic lot scheduling problem (ELSP) deals with scheduling of multiple products in a single given machine to minimize total cost by determining both a production sequence and a lot size for each product. It is typically assumed that demand rates and setup times are known. ELSP which is an NP hard problem [1] has been studied by a large number of researchers for more than 50 years. The methodologies developed are based on any one of the common cycle, basic period, or time-varying lot size approaches. The common cycle approach is the simplest to implement when all products are manufactured in the same period while the basic period approach allows different cycle times for different products. However, cycle times must be an integer multiple of a basic period. Unlike the first two approaches, the time-varying lot size approach is more flexible than the other two approaches since lot sizes may vary for different in one cycle [2, 3]. Recently, meta-heuristics such as simulated annealing (SA) [4, 5], genetic algorithm (GA) [6, 7, 8, 9], tabu search (TS) [10] etc. have motivated the time-varying lot size approach to solve the problem. In this study, since the traditional deterministic ELSP models neglect the uncertainty of the demand rates and setup times and so may perform poorly under certain realizations of the random demand rates and setup times, we exploit the assumption of deterministic demand rates and setup times and formulate the problem as a bi-objective optimization problem that aims at providing solution robust production schedules. Based on the solution approach presented in Goncalves and Sousa [9] which propose a GA combined with linear programming (LP) that considers initial and ending inventory levels as given and allows backorders, we adopt a hybrid approach, combining a genetic algorithm and linear programming. The aim of this approach is to generate a set of non-dominated solution robust production schedules, i.e., production schedules that do not differ much from the actually realized schedules in terms of makespan and total cost. The details of the solution approach are presented in Section 2. In Section 3, the details of the experimental analysis are provided. Finally in Section 4, the paper is concluded with a discussion on the results of the case study, final remarks and possible future research agenda.

2 Solution Approach

In this section we present a solution approach combining a bi-objective GA with linear programming based on the formulation presented in Goncalves and Sousa [9]. The only difference is in the objective functions. While Goncalves and Sousa [9] considers a single objective (total cost), in the model utilized in this paper, the aim of the proposed hybrid approach is to generate non-dominated solution robust production schedules by considering the uncertainty in the demand rates and setup times. Solution robustness aims at constructing a production schedule that differs from the realized production schedule in the least possible amount. Therefore, in our solution approach we aim to minimize the difference between the makespan and total cost of the production schedule and actually realized makespan and total cost. This difference between the production schedule and the actually realized schedule is measured with the total sum of absolute deviations (TSAD) for both makespan and average cost of the schedule through K number of possible schedule realizations. Therefore, the objectives considered in the bi-objective GA are to minimize the average TSAD of makespan $TSAD_{avg}^{makespan}$ and average TSAD of average cost $TSAD_{avg}^{avgcost}$. Proposed bi-objective GA is an adopted version of NSGA-II [11] and GA presented by Goncalves and Sousa [9]. Chromosome representation, chromosome decoding procedure and schedule generation scheme are exactly the same as presented in Goncalves and Sousa [9] and population management is the same as of NSGA-II. However, our bi-objective GA differs in the chromosome evaluation procedure. Initial population is comprised of randomly generated chromosomes and we make use of one-point crossover and swap mutation operators. Therefore, in the following subsections, we present only the chromosome evaluation procedure.

Evaluation for a chromosome is based on a set of K realizations reflecting the uncertainty around the demand rates and setup times. For a given chromosome both the $TSAD_{avg}^{makespan}$ and $TSAD_{avg}^{avgcost}$ are assessed through a set of K realizations mimicking the implementation phase, where a realization corresponds to a sample instance obtained by a simulation run using the demand rate and setup time distributions. To calculate $TSAD_{avg}^{makespan}$ and $TSAD_{avg}^{avgcost}$ of a chromosome, first a set of K realizations are performed. For each such realization, production schedule is obtained with the schedule generation scheme explained in the previous section. Hence, K production schedules each having its own makespan and average cost are obtained. Then, using the formula given in Equation (1) and Equation (2), for every realization of k, $TSAD_{makespan}^{k}$ and $TSAD_{avgcost}^{k}$ are calculated.

$$TSAD_{makespan}^{k} = \sum_{r \in R: r \neq k} |makespan_k - makespan_r| \tag{1}$$

$$TSAD_{avgcost}^{k} = \sum_{r \in R: r \neq k} |avgcost_k - avgcost_r| \tag{2}$$

R is the set of realizations and r is the realization index. These K realizations are then sorted in their non-domination levels using the corresponding $TSAD$ values and the schedules that have a rank value of 1 constitute the robust non-dominated schedule set of the chromosome. Fitness pair of a chromosome ($TSAD_{avg}^{makespan}$,$TSAD_{avg}^{avgcost}$) is then simply calculated by taking the averages of fitness pairs of schedules in the non-dominated schedule set of the chromosome.

3 Experimental Analysis

In this section, we report the results obtained on a set of experiments conducted to evaluate the performance of the bi-objective GA. In this paper we only present the experimental results of a pilot study. In this pilot study, we assume that demand rates and setup times are known in advance. Consequently, we minimize makespan and average cost of production schedules. Following the same settings as in Goncalves and Sousa [9], we also impose that the initial inventory be equal to the final inventory to force the linear programming model to produce a cycle. Analysis is conducted by generating two sets of problem instances each with 5 randomly generated problems using the uniform distribution for the parameters given in Table 1.

3.1 Fine-Tuning of the GA Parameters

Since the parameters used in GAs have a direct effect on the performance, there is a need for a fine-tuning procedure to select the best parameter combinations to be used in the implementation of the proposed solution approach. The best combination

	# products (units)	Production Rate (units/units time)	Demand Rate (units/units time)	Setup Time (time)	Setup Cost ($)	Holding Cost ($)
Set1	5	[1500-5000]	[150-1200]	[0.10-0.40]	[40-160]	[0.003-0.010]
Set2	10	[1500-5000]	[150-1200]	[0.10-0.40]	[40-160]	[0.003-0.010]

Table 1: Problem parameter ranges for randomly generated test instances

of the parameters to be used in the bi-objective GA are determined through experimentation. For this experiment, two problem instances from each set is selected and solved with the bi-objective GA. Using each value of the GA parameters that we tested crossover rate [0.60, 0.75], population size [50,100], mutation rate [0.10,0.15], population size [50,100], number of generations [30,50]) a total of 24 parameter combinations is obtained and tested by solving each of the determined test instances three times to reduce the undesired effect of randomness. Thus, we have run our bi-objective four times for each project-parameter combinations, yielding a total of 288 runs.

To compare the performances of the parameter combinations we have used the following four different performance measures: Scaled Extreme Hyperarea Ratio (SEHR), Maximum Spread (MS), Number of Non-dominated Solutions (#ND), CPU Time (CPU). Scaled extreme hyperarea ratio (Scaled EHR) that we use to compare the performances of different GA parameter combinations is based on the hypervolume indicator [12] and extreme hyperarea ratio (EHR) [13] and it measures the quality of the solutions. The maximum spread metric (MS) [12] shows how much the non-dominated solutions in the non-dominated front spread. The larger the MS value, the better the spread the corresponding parameter combination achieves. To summarize, we prefer parameter combinations with smaller SEHR values, higher MS values, higher #ND and smaller CPU. To determine the best parameter combinations, non-dominated sorting is applied, then, among the non-dominated performance quadruples the one with the largest weighted scores is chosen with its corresponding parameter combination being the one to be implemented in the computational study to follow.

The results show that the best parameter combination with normalized performance measure values and weighted score of 0.81 for these parameter combinations over the tested 24 parameter combinations obtained after the fine tuning procedure is with crossover rate as 0.90, mutation rate as 0.10, population size as 50, and the number of generations as 30. Therefore in the experimental analysis, we make use of these parameter combinations to obtain the robust production schedules.

3.2 Computational Results

Our bi-objective GA is implemented in C# and the computational experiments were carried out on a computer with an Intel i5 1.80 GHz CPU and 12 GB RAM. Table 2 shows the best average cost, best makespan among the non-dominated solutions of each problem instance with the CPU times and the number of non-dominated solutions.

Problem No	Problem Size	Best Avg. Cost	Best Makespan	CPU Time (seconds)	Non-dominated Solution Count
1	5	209.19	5.59	8.21	1
2	5	1031.96	1.72	9.56	1
3	5	303.55	1.49	10.06	2
4	5	402.17	1.64	10.26	1
5	5	515.66	11.87	7.82	1
6	10	975.39	3.11	34.36	4
7	10	854.81	3.28	35.84	9
8	10	564.27	3.03	44.4	3
9	10	1375.22	3.24	40.72	8
10	10	927.58	3.16	41.86	3

Table 2: Performance results for the problem instances

4 Conclusion

In this paper, we have addressed the problem of scheduling economic lots in a multi-product single-machine environment and formulated it as a multi-objective optimization problem. Demand rates and setup times are taken as random variables from a known distribution. To solve the problem, we developed a hybrid approach combining bi-objective GA and LP. Minimization of expected total cost and expected makespan is aimed to generate solution robust production schedules. These expectations are calculated through K number of schedule realizations mimicking the implementation phase where a realization corresponds to a sample instance obtained by a simulation run using the demand rate and setup time distributions. A pilot study is done for assuming the demand rates and setup times are known in advance with the objective of minimizing the makespan and average cost. Results of this pilot study are presented. Further research might be directed for the case where there are multiple identical machines.

References

[1] Hsu, W. L., On the general feasibility test of scheduling lot sizes for several products on one machine, Management Science, 29(1), 93-105 (1983)

[2] Maxwell, W. L., The scheduling of economic lot sizes, Naval Research Logistics Quarterly, 11(2), 89-124 (1964)

[3] Delporte, C. M., and Thomas, L. J, Lot sizing and sequencing for N products on one facility, Management Science, 23(10), 1070-1079 (1977)

[4] Feldmann, M., and Biskup, D., Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches, Computers & Industrial Engineering, 44(2), 307-323 (2003).

[5] Raza, A. S., and Akgunduz, A., A comparative study of heuristic algorithms on economic lot scheduling problem, Computers & Industrial Engineering, 55(1), 94-109 (2008).

[6] Khouja, M., Michalewicz, Z., and Wilmot, M., The use of genetic algorithms to solve the economic lot size scheduling problem, European Journal of Operational Research, 110(3), 509-524 (1998)

[7] Moon, I., Silver, E. A., and Choi, S., Hybrid genetic algorithm for the economic lot-scheduling problem, International Journal of Production Research, 40(4), 809-824 (2002)

[8] Gaafar, L., Applying genetic algorithms to dynamic lot sizing with batch ordering, Computers & Industrial Engineering, 51(3), 433-444 (2006)

[9] Goncalves, J. F., and Sousa, P. S., A genetic algorithm for lot sizing and scheduling under capacity constraints and allowing backorders, International Journal of Production Research, 49(9), 2683-2703 (2011)

[10] Raza, S. A., Akgunduz, A., and Chen, M. Y., A tabu search algorithm for solving economic lot scheduling problem, Journal of Heuristics, 12(6), 413-426 (2006)

[11] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A. M. T., A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, 6(2), 182-197 (2002)

[12] Zitzler, E., and Thiele, L., Multi-objective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation, 3(4), 257-271 (1999)

[13] Kilic, M., Ulusoy, G., Serifoglu, F. S., A bi-objective genetic algorithm approach to risk mitigation in project scheduling, International Journal of Production Economics, 112(1), 202-216 (2008)

A Theoretical Study of Two-Period Relaxations for Lot-Sizing Problems with Big-Bucket Capacities

Kerem Akartunalı
Dept. of Management Science, University of Strathclyde, Glasgow, UK
kerem.akartunali@strath.ac.uk

Mahdi Doostmohammadi
Dept. of Management Science, University of Strathclyde, Glasgow, UK
mahdi.doostmohammadi@strath.ac.uk

Ioannis Fragkos
HEC Montréal, Canada
ioannis.fragkos@hec.ca

Abstract

In this paper, we study two-period subproblems proposed by [1] for lot-sizing problems with big-bucket capacities and nonzero setup times, complementing our previous work [3] investigating the special case of zero setup times. In particular, we study the polyhedral structure of the mixed integer sets related to various two-period relaxations. We derive several families of valid inequalities and investigate their facet-defining conditions. We also discuss the separation problems associated with these valid inequalities.

1 Introduction

In this study, we investigate multi-item production planning problems with big bucket capacities, i.e., each resource is shared by multiple items, which can be produced in a specific time period. These real-world problems are very interesting, as they remain challenging to solve to optimality and also to achieve strong bounds. The uncapacitated and single-item relaxations of the problem have been previously studied by [7]. The work of [6] introduced and studied the single-period relaxation with "preceding inventory", where a number of cover and reverse cover inequalities are defined for this relaxation. Finally, we also note the relevant study of [5], who studied a single-period relaxation and compared with other relaxations.

We present a formulation for this problem following the notation of [2]. Let NT, NI and NK indicate the number of *periods*, *items*, and *machine types*, respectively.

We represent the production, setup, and inventory variables for item i in period t by x_t^i, y_t^i, and s_t^i, respectively. We note that our model can be generalized to involve multiple levels as in [1], however, we omit this for the sake of simplicity.

$$\min \sum_{t=1}^{NT}\sum_{i=1}^{NI} f_t^i y_t^i + \sum_{t=1}^{NT}\sum_{i=1}^{NI} h_t^i s_t^i \tag{1}$$

$$\text{s.t. } x_t^i + s_{t-1}^i - s_t^i = d_t^i \qquad t \in \{1,\ldots,NT\}, i \in \{1,\ldots,NI\} \tag{2}$$

$$\sum_{i=1}^{NI}(a_k^i x_t^i + ST_k^i y_t^i) \leq C_t^k \qquad t \in \{1,\ldots,NT\}, k \in \{1,\ldots,NK\} \tag{3}$$

$$x_t^i \leq M_t^i y_t^i \qquad t \in \{1,\ldots,NT\}, i \in \{1,\ldots,NI\} \tag{4}$$

$$y \in \{0,1\}^{NT\text{x}NI}; x, s \geq 0 \tag{5}$$

The objective function (1) minimizes total cost, where f_t^i and h_t^i indicate the setup and inventory cost coefficients, respectively. The flow balance constraints (2) ensure that the demand for each item i in period t, denoted by d_t^i, is satisfied. The big bucket capacity constraints (3) ensure that the capacity C_t^k of machine k is not exceeded in time period t, where a_k^i and ST_k^i represent the per unit production time and setup time for item i, respectively. The constraints (4) guarantee that the setup variable is equal to 1 if production occurs, where M_t^i represents the maximum number of item i that can be produced in period t, based on the minimum of remaining cumulative demand and capacity available. Finally, the integrality and non-negativity constraints are given by (5).

2 Two-Period Relaxation

Let $I = \{1,\ldots,NI\}$. We present the feasible region of a two-period, single-machine relaxation of the multi-item production planning problem, denoted by X^{2PL} (see [1] for details).

$$x_{t'}^i \leq \widetilde{M}_{t'}^i y_{t'}^i \qquad i \in I, t' = 1,2 \tag{6}$$

$$x_{t'}^i \leq \widetilde{d}_{t'}^i y_{t'}^i + s^i \qquad i \in I, t' = 1,2 \tag{7}$$

$$x_1^i + x_2^i \leq \widetilde{d}_1^i y_1^i + \widetilde{d}_2^i y_2^i + s^i \qquad i \in I \tag{8}$$

$$x_1^i + x_2^i \leq \widetilde{d}_1^i + s^i \qquad i \in I \tag{9}$$

$$\sum_{i \in I}(a^i x_{t'}^i + ST^i y_{t'}^i) \leq \widetilde{C}_{t'} \qquad t' = 1,2 \tag{10}$$

$$x, s \geq 0, y \in \{0,1\}^{2\times NI} \tag{11}$$

Since we consider a single machine, we dropped the k index from this formulation, however, all parameters are defined in the same lines as before. The obvious choice for the horizon would be $t+1$, in which case the definition of the parameter $\widetilde{M}^i_{t'}$ is the same as of the basic definition of $M^i_{t+t'-1}$, for all i and $t' = 1,2$. Similarly, capacity parameter $\widetilde{C}_{t'}$ is the same as $C_{t+t'-1}$, for all $t' = 1,2$. Cumulative demand parameter $\widetilde{d}^i_{t'}$ represents simply $d^i_{t+t'-1,\, t+1}$, for all i and $t' = 1,2$, i.e., $\widetilde{d}^i_1 = d^i_{1,2}$ and $\widetilde{d}^i_2 = d^i_2$. We note the following polyhedral result for X^{2PL} from [1].

Proposition 2.1 *Assume that $\widetilde{M}^i_t > 0, \forall t \in \{1,\dots,NT\}, \forall i \in \{1,\dots,NI\}$ and $ST^i < \widetilde{C}_t, \forall t \in \{1,\dots,NT\}, \forall i \in \{1,\dots,NI\}$. Then conv($X^{2PL}$) is full-dimensional.*

For the sake of simplicity, we will drop subscript t and symbol $\widetilde{}$ in the following notations. In this paper, we investigate the case of $a^i = 1, \forall i \in \{1,\dots,NI\}$ with nonzero setups. We establish two relaxations of X^{2PL} and study their polyhedral structures. For a given t, we define the first relaxation of X^{2PL}, denoted by $LR1$, as set of $(x,y) \in \mathbb{R}^{NI} \times \mathbb{Z}^{NI}$ satisfying

$$
\begin{aligned}
& x^i \le M^i y^i, i \in I \\
& \sum_{i=1}^{NI} (x^i + ST^i y^i) \le C \\
& x^i \ge 0, y^i \in \{0,1\}, i \in I
\end{aligned}
$$

Next, we present a result from the literature [4] concerning this relaxation.

Definition 2.1 *Let $S_1 \subseteq I$ and $S_2 \subseteq I$ such that $S_1 \cap S_2 = \emptyset$. We say that (S_1, S_2) is a* generalized cover *of I if $\sum_{i\in S_1}(M^i + ST^i) + \sum_{i \in S_2} ST^i - C = \delta > 0$.*

Proposition 2.2 *(see [4]) Let (S_1,S_2) be a generalized cover of I, and let $L_1 \subseteq I \backslash (S_1 \cup S_2)$ and $L_2 \subseteq I \backslash (S_1 \cup S_2)$ such that $L_1 \cap L_2 = \emptyset$. Then,*

$$
\begin{aligned}
\sum_{i\in S_1\cup L_1} x^i + \sum_{i \in S_1\cup S_2\cup L_1 \cup L_2} ST^i y^i - \sum_{i\in S_1}(M^i + ST^i - \delta)^+ y^i - \sum_{i\in S_2}(ST^i-\delta)^+ y^i \\
- \sum_{i\in L_1}(\overline{q}^i - \delta) y^i - \sum_{i \in L_2}(\overline{ST}^i - \delta) y^i \le C - \sum_{i\in S_1}(M^i + ST^i - \delta)^+ - \sum_{i\in S_2}(ST^i - \delta)^+
\end{aligned}
$$

is valid for $LR1$, where $A \ge \max(\max_{i\in S_1}(M^i + ST^i), \max_{i\in S_2} ST^i, \delta)$, $\overline{q}^i = \max(A, M^i + ST^i)$, and $\overline{ST}^i = \max(A, ST^i)$.

For a given t, second relaxation of X^{2PL}, denoted by $LR2$, can be defined as the set of $(x, y, s) \in \mathbb{R}^{NI} \times \mathbb{Z}^{NI} \times \mathbb{R}^{NI}$ satisfying

$$
\begin{aligned}
& x^i \leq M^i y^i, i \in I \\
& x^i \leq d^i y^i + s^i, i \in I \\
& \sum_{i=1}^{NI} (x^i + ST^i y^i) \leq C \\
& x^i \geq 0, y^i \in \{0, 1\}, s^i \geq 0, i \in I
\end{aligned}
$$

In this talk, we will present the trivial facet-defining inequalities for $LR2$, and then derive several classes of valid inequalities such as *cover* and *partition* inequalities. We will also present item- and period-extended versions of some of these families of inequalities, and we will establish facet-defining conditions for all families of inequalities. We will also discuss the separation problems associated with these valid inequalities.

References

[1] K. Akartunalı, I. Fragkos, A.J. Miller and T. Wu, Local cuts and two-period covex hull closures for big bucket lot-sizing problems, Technical Report, Dept. of Management Science, University of Strathclyde, Available at Optimization Online, `http://www.optimization-online.org/DB_HTML/2014/07/4423.html` (2014).

[2] K. Akartunalı and A.J. Miller, A computational analysis of lower bounds for big bucket production planning problems, Computational Optimization and Applications, 53, 729-753 (2012).

[3] M. Doostmohammadi and K. Akartunalı, A Polyhedral Study of Two-Period Relaxations for Big-Bucket Lot-Sizing Problems: Zero Setup Case, Technical Report, Dept. of Management Science, University of Strathclyde, Available at Optimization Online, `http://www.optimization-online.org/DB_HTML/2015/02/4767.html` (2015).

[4] M. Goemans, Valid ineqialites and separation for mixed 0-1 constraints with variable upper bounds, Operations Research Letters, 8, 315-322 (1989).

[5] R. Jans and Z. Degraeve, Improved lower bounds for the capacitated lot sizing problem with setup times, Operations Research Letters, 32, 185-195 (2004).

[6] A.J. Miller, G.L. Nemhauser and M.W.P. Savelsbergh, Solving multi-item capacitated lot-sizing problems with setup times by branch-and-cut, Technical report CORE DP 2000/39, Université Catholique de Louvain, Louvain-la-Neuve (2000).

[7] Y. Pochet, Valid inequalities and separation for capacitated economic lot-sizing, Operations Research Letters, 7, 109-116 (1988).

A Computational Study of Two-Period Relaxations for Lot-Sizing Problems with Big-Bucket Capacities

Ioannis Fragkos
HEC Montréal, Canada
i.fragkos@hec.ca

Kerem Akartunalı
Dept. of Management Science, University of Strathclyde, Glasgow, UK
kerem.akartunali@strath.ac.uk

Mahdi Doostmohammadi
Dept. of Management Science, University of Strathclyde, Glasgow, UK
mahdi.doostmohammadi@strath.ac.uk

Abstract

The multi-item Capacitated Lot-sizing problem with Setup Times (CLST) is an important problem from both a theoretical and a practical perspective. This talk is part of a research stream that studies two-period relaxations of CLST. We present computational experiments that investigate the strength of valid inequalities that are derived from two-period relaxations. Four families of valid inequalities are considered, all of which are generalisations of cover inequalities, as described in Padberg et al., [4]. We present a numerical study in which we compare the strength of these inequalities with the (l, S) inequalities of Barany et al. [2]. We find that, for certain instances, some families are very efficient, and are able to improve the lower bound by a great margin.

1 Introduction

The multi-item Capacitated Lot-sizing problem with Setup Times (CLST) problem asks for the production decisions that minimise the joint setup and inventory costs of a given set of items over a given time horizon. Each item should satisfy a demand quantity in each period, and, whenever it is produced at a positive level, a machine setups is required. A machine can produce multiple items within a time period, while machine setups consume production time, which is limited, and also incur known costs. Excess production can be carried over to satisfy the demand of subsequent periods but cannot be used to cover the demand of past periods. The mathematical formulation of the problem makes use of the following quantities.

Parameters
NT: Number of time periods
NI: Number of items
NK: Number of machines
f_t^i: Setup cost of item i in period t
h_t^i: Inventory cost for carrying one unit of item i through period t
d_t^i: Demand of item i in period t
ST_k^i: Setup time required for item i in machine k
α_k^i: Unit production time of item i in machine k
C_t^k: Capacity of machine k in time period t

We also define the maximum allowable production quantity of item i in period t as $M_t^i = \min\{\sum_{l=t}^{NT} d_l^i, \min_k \frac{C_t^k - ST^i}{\alpha_k^i}\}$

Decision Variables
$x_t^i \geq 0$: Production quantity of item i in period t
$s_t^i \geq 0$: Inventory quantity of item i at the beginning of period t
$y_t^i \in \{0, 1\}$: One when a setup is scheduled for item i in period t, zero otherwise

The problem can then be formulated as follows.

$$\min \sum_{t=1}^{NT} \sum_{i=1}^{NI} f_t^i y_t^i + \sum_{t=1}^{NT} \sum_{i=1}^{NI} h_t^i s_t^i \tag{1}$$

$$\text{s.t. } x_t^i + s_{t-1}^i - s_t^i = d_t^i \qquad t \in \{1, \ldots, NT\}, i \in \{1, \ldots, NI\} \tag{2}$$

$$\sum_{i=1}^{NI} (a_k^i x_t^i + ST_k^i y_t^i) \leq C_t^k \qquad t \in \{1, \ldots, NT\}, k \in \{1, \ldots, NK\} \tag{3}$$

$$x_t^i \leq M_t^i y_t^i \qquad t \in \{1, \ldots, NT\}, i \in \{1, \ldots, NI\} \tag{4}$$

$$y \in \{0, 1\}^{NT\text{x}NI}; x, s \geq 0 \tag{5}$$

The objective function (1) minimizes total cost, that consists of the joint cost of setups and inventory holdings, respectively. The flow balance constraints (2) ensure that the demand for each item i in period t is satisfied. The big bucket capacity constraints (3) ensure that the capacity C_t^k of machine k is not exceeded in time period t, while constraints (4) guarantee that no production occurs unless a setup is made. Finally, constraints (5) describe the non-negativity and integrality restrictions.

2 Two-Period Relaxation

Let $I = \{1, \dots, NI\}$. We present the feasible region of a two-period, single-machine relaxation of the multi-item production planning problem, denoted by X^{2PL}. This relaxation is introduced in Akartunalı et al., [1], and a polyhedral study for the special case of zero setup costs is presented in Doostmohammadi and Akartunalı, [3].

$$x^i_{t'} \leq \widetilde{M}^i_{t'} y^i_{t'} \qquad i \in I, t' = 1,2 \tag{6}$$

$$x^i_{t'} \leq \widetilde{d}^i_{t'} y^i_{t'} + s^i \qquad i \in I, t' = 1,2 \tag{7}$$

$$x^i_1 + x^i_2 \leq \widetilde{d}^i_1 y^i_1 + \widetilde{d}^i_2 y^i_2 + s^i \qquad i \in I \tag{8}$$

$$x^i_1 + x^i_2 \leq \widetilde{d}^i_1 + s^i \qquad i \in I \tag{9}$$

$$\sum_{i \in I} (a^i x^i_{t'} + ST^i y^i_{t'}) \leq \widetilde{C}_{t'} \qquad t' = 1,2 \tag{10}$$

$$x, s \geq 0, y \in \{0,1\}^{2 \times NI} \tag{11}$$

Since we consider a single machine, we dropped the k index from this formulation, however, all parameters are defined in the same lines as before. The obvious choice for the horizon would be $t+1$, in which case the definition of the parameter $\widetilde{M}^i_{t'}$ is the same as of the basic definition of $M^i_{t+t'-1}$, for all i and $t' = 1,2$. Similarly, capacity parameter $\widetilde{C}_{t'}$ is the same as $C_{t+t'-1}$, for all $t' = 1,2$. Cumulative demand parameter $\widetilde{d}^i_{t'}$ represents simply $d^i_{t+t'-1,\, t+1}$, for all i and $t' = 1,2$, i.e., $\widetilde{d}^i_1 = d^i_{1,2}$ and $\widetilde{d}^i_2 = d^i_2$.

3 Preliminary Results

We employ the instances of Trigeiro et al. as modified by Sural et al., [5]. It is worth noticing that the elimination of setup costs increases the integrality gaps of those instances from 0.97% to 33.86%, where the lower bound is calculated by Lagrange relaxation and is equivalent to that obtained by introducing the (l, S) inequalities at formulation (1)-(5). Table 1 reports on the improvement of the integrality gap obtained by separating two new families of new inequalities (two-period covers and partition I) in addition to (l, S) inequalities.

The talk will also discuss the separation problem for each family of inequalities and the challenges in the design of an embedded branch-and-cut framework.

References

[1] K. Akartunalı, I. Fragkos, A.J. Miller and T. Wu, Local cuts and two-period covex hull closures for big bucket lot-sizing problems, Technical Report, Dept.

of Management Science, University of Strathclyde, Available at Optimization Online, http://www.optimization-online.org/DB_HTML/2014/07/4423.html (2014).

[2] I. Barany, T.J. Van Roy and L.A. Wolsey, Uncapacitated lot-sizing: The convex hull of solutions, *Mathematical Programming Studies*, 22, 32-43 (1984).

[3] M. Doostmohammadi and K. Akartunalı, A Polyhedral Study of Two-Period Relaxations for Big-Bucket Lot-Sizing Problems: Zero Setup Case, Technical Report, Dept. of Management Science, University of Strathclyde, Available at Optimization Online, http://www.optimization-online.org/DB_HTML/2015/02/4767.html (2015).

[4] M.W. Padberg, T.J. Van Roy and L.A. Wolsey, Valid linear inequalities for fixed charge problems,*Operations Research*, 33, 842-861 (1985).

[5] H. Süral, M. Denizel and L.N. Van Wassenhove, Lagrangean relaxation based heuristics for lot sizing with setup times, *European Journal of Operational Research*, 194, 51-63 (2009).

[6] W. Trigeiro, L.J. Thomas and J. O. McClain, Capacitated lot sizing with setup times, *Management Science*, 24, 353-366 (1989).

Instance	LP Gap	LS Gap	LS+2PC+PI Gap
G1	99%	99%	48%
G10	0%	0%	0%
G11	89%	89%	28%
G13	74%	19%	10%
G14	77%	13%	3%
G16	60%	24%	13%
G17	55%	10%	9%
G18	43%	5%	4%
G19	82%	22%	13%
G2	100%	99%	59%
G20	56%	17%	9%
G21	97%	97%	96%
G22	98%	98%	98%
G23	98%	98%	98%
G24	99%	13%	13%
G25	98%	97%	97%
G26	99%	99%	99%
G27	94%	94%	94%
G28	99%	47%	47%
G29	99%	99%	98%
G3	100%	100%	58%
G30	99%	99%	99%
Average	82%	59%	47%

Table 1: Improvement of gap with the addition of two-period cover (2PC) and Partition-I (PI) inequalities.

Index of Authors