International Workshop on Lot-Sizing - IWLS'2023

23th-25th August 2023, Cork, Ireland

Table of contents

Workshop Organization

Local organizing Committee

- Andrea Visentin, Insight Centre for Data Analytics, Ireland.
- Barry O'Sullivan, Insight Centre for Data Analytics, Ireland.
- S. Armagan Tarim, Cork University Business School, Ireland.
- Roberto Rossi, University of Edinburgh Business School, UK.
- Linda O'Sullivan, Insight Centre for Data Analytics, Ireland.
- Caitriona Walsh, Insight Centre for Data Analytics, Ireland.
- Eleanor O'Riordan, Insight Centre for Data Analytics, Ireland.

Steering Committee

- Nabil Absi (France)
- Kerem Akartunali (UK)
- Bernardo Almada-Lobo (Portugal)
- Christian Almeder (Germany)
- Stéphane Dauzère-Pérès (France)
- Silvio de Araujo (Brasil)
- Meltem Denizel (Turkey)
- Raf Jans (Canada)
- Hark-Chin Hwang (South Korea)
- Safia Kedad-Sidhoum (France)
- Edwin Romeijn (USA)
- Florian Sahling (Germany)
- Wilco van den Heuvel (Netherlands)
- Laurence A. Wolsey (Belgium)

List of sponsors

EURO working group on Lot Sizing

Insight Centre for Data Analytics

University College Cork
School of Computer Science & IT

EURO
EURO is the 'Association of European Operational Research Societies'

TAILOR
A Network for Trustworthy Artificial Intelligence

Time	Tuesday 22/08	Wednesday 23/08 Aula Maxima	Thursday 24/08 WGB 1.07	Friday 25/08 WGB 1.07
8:30 - 9:00		Registration		
9:00 - 9:30		Opening Session	Session 4 - Multi-level Manufacturing	Session 7 - Mathematical Formulation and Reformulations
9:30 - 10:00		Session 1 - Cutting Stock		
10:00 - 10:30				
10:30 - 11:00		Break	Break	Break
11:00 - 11:30		Session 2 - Sustainability	Session 5 - Inventory Control	Session 8 - Remanufacturing
11:30 - 12:00				
12:00 - 12:30				
12:30 - 13:00		Lunch	Lunch	Lunch
13:00 - 13:30				
13:30 - 14:00				
14:00 - 14:30		Session 3 - Energy	A word from Insight	
14:30 - 15:00			Session 6 - Machine Learning	
15:00 - 15:30				
15:30 - 16:00		Break	Break	
16:00 - 16:30		Guided vist of the Campus	Meeting EURO WGLT	
16:30 - 17:00				
17:00 - 17:30				
17:30 - 18:00				
18:00 - 18:30				
18:30 - 19:00	Get Together Party Mardyke Entertainment Complex			
19:00 - 19:30		Pizza Dinner Franciscan Well Brewery	Gala Dinner Isaacs Restaurant	
19:30 - 22:30				

Sessions Breakdown

Session 1 **Cutting Stock** - (chair: Stéphane Dauzère-Pérès)

A simultaneous lot-sizing, sequencing and cutting stock problems in the furniture industry
Maissa Mati, Abderrahim Sahli, <u>Sana Belmokhtar-Berraf</u>

A 3-level integrated lot-sizing and cutting stock problem with supplier selection and distribution decisions
<u>Silvio Alexandre de Araujo</u>, Gislaine Mara Melega, Raf Jans, Reinaldo Morabito

Session 2 **Sustainability** - (chair: Roberto Rossi)

A Heuristic Algorithm to solve the One-Warehouse Multi-Retailer Problem with an Emission Constraint
<u>Matthieu Gruson</u>, Raf Jans, Ola Jabali, Quihua Zhong

Coordinating shipments in lot-sizing models
<u>Wilco van den Heuvel</u>, Marcel Turkensteen, Rommert Dekker

Network Design for Closed-Loop Supply Chain Network with Hybrid Retailers/Collection Centres
<u>Mahdi Doostmohammadi, Mehdi Amiri-Aref</u>

Session 3 **Energy** - (chair: Raf Jans)

Ensuring fair allocation of renewable energy in microgrids for supply planning
Natalia Jorquera-Bravo, Sourour Elloumi, <u>Safia Kedad-Sidhoum</u>, Agnnes Plateau

First results on energy-oriented lot-sizing and scheduling with energy storage
<u>Stephan Köppel</u>, Florian Sahling

A two-stage stochastic programming model for lot-sizing with onsite generation of renewable energy
Ruiwen Liao, Franco Quezada, Cèline Gicquel, Safia Kedad-Sidhoum

Session 4	**Multi-level Manufacturing** - (chair: Silvio A. de Araujo)

Integrated shelf-life rules for multi-level tablets manufacturing processes
Michael Simonis, Stefan Nickel

MILP-based local search procedures for minimizing total tardiness in the No-idle Permutation Flowshop Problem
Andrea Balogh, Michele Garraffa, Barry O'Sullivan, Fabio Salassa

A real-life batch-sizing and sequencing problem with buffer & sequence-dependent setup times
Nicola La Palombara, Walid Behiri, Sana Belmokhtar-Berraf, Chengbin Chu, Mustapha Sali

Session 5	**Inventory Control** - (chair: Wilco van den Heuvel)

Inventory Analytics
Roberto Rossi

Stochastic Dynamic Programming heuristic for the (R, s, S) policy parameters computation
Andrea Visentin, Steven Prestwich, Roberto Rossi, S Armagan Tarim

A Heuristic Method for Perishable Inventory Management Under Non-Stationary Demand
Suheyl Gulecyuz, Barry O'Sullivan, S. Armagan Tarim

Session 6	**Machine Learning** - (chair: Andrea Visentin)

Learning capacity consumption in multi-level capacitated lot-sizing problems
David Tremblet, Alexandre Dolgui, Simon Thevenin

A machine learning approach for identifying the best solution heuristic for a large scaled Capacitated Lotsizing Problem
Jens Kärcher, Herbert Meyr

Session 7	**Mathematical Formulation and Reformulations** - (chair: Steven Prestwich)

Formulations for the one-warehouse multi-retailer problem with production constraints
Agathe L'Hermite, Marilène Cherkesly, Matthieu Gruson

Timed Route Approaches for Production Planning with Time Constraints
Benjamin Anthouard, Quentin Christ, Stephane Dauzere-Peres, Renaud Roussel

The integrated lot-sizing and storage assignment problem
Gislaine Mara Melega, Chi Xu, Raf Jans, Julie Paquette

Session 8	**Remanufacturing** - (chair: Safia Kedad-Sidhoum)

Robust Models for Remanufacturing with Multiple Quality Classes
Farzad Sham, Agostinho Agra, Kerem Akartunali

Uncapacitated lot-sizing problems with remanufacturing and two sale markets
Siao-Leu Phouratsamay, Bernard Penz

First results on integrated stochastic lot sizing and rework planning
Pierre Kohlmann, Florian Sahling

Cutting Stock

Aula Maxima - Wednesday, 23/08 - 09:30-10:30

A simultaneous lot-sizing, sequencing and cutting stock problems in the furniture industry

Maissa Mati, Abderrahim Sahli and Sana Belmokhtar-Berraf
Grettia, Université Gustave Eiffel, 16 Bd Newton, Champs sur Marne 77420
maissa.mati, abderrahim.sahli, sana.berraf-belmokhtar@univ-eiffel.fr

Abstract

This study addresses the integration of lot-sizing, sequencing, and cutting stock problems in the furniture industry where the production system is composed of two stages with a buffer in between. The buffer is filled in the first stage with pieces cut from wood panels to ensure a continuous painting process in the second phase, which is the bottleneck. The planning issue is to determine the total number of wood panels to be cut and the quantities of pieces to be painted using a robotic arm, as well as their sequence, in order to meet demand within a finite time horizon while minimizing inventory and setup costs. The study proposes several mixed-integer programming models that are compared based on randomly generated instances based on real data.

1 Introduction

The lot-sizing and cutting-stock problems have been extensively studied for over 50 years [1]. Significant progress has been made in modeling and solving each problem individually, considering their NP-hard nature. However, recent interest has grown in integrating these problems due to advancements in optimization theory, hardware, software, and observed interdependencies across diverse industries [1]. This integration aims to consider both decisions simultaneously, capturing their interdependence for a global optimization. The authors of [2] recently investigated the integration of these two problems in a study focused on the manufacturing process of aerospace composite products. A classification of literature on the integration of lot-sizing and cutting-stock problems can be found in [1]. This classification revolves around two main types of integration: integration across multiple periods of the planning horizon through inventory management and integration between different production levels, including bin provisioning, cutting, and production of finished products. Our problem considers a multi-period, multi-level integration approach, excluding the first level which involves the supply of wood panels. Several papers, including [4, 3] and others, are cited in [1], addressing similar integration types to our study.

2 Problem model

We have proposed three MIP models for the considered problem. Here, we present the most effective one, which determines the required quantities of bins and items to be cut in each period, as well as the coloring batches and color sequence for each period. Let $\mathcal{T}$ be the set of periods, $\mathcal{B}$ be the set of bins, $\mathcal{J}$ be the set of item types, and $\mathcal{C}$ be the set of colors. For each period t, the demand for items of type j colored in c is denoted as d_{jct} and is known in advance. The duration of each period t is denoted as f_t. Each item of type j can only be cut from a compatible bin b. The binary parameter g_{jb} indicates the compatibility of item j with bin b. To simplify the problem, colors are grouped into three categories: light, medium, and dark, along with a fictitious color denoted as 0, which represents the cleaned state of the robot. This color serves as the initial and final color in each period. Let $\mathcal{M}$ be the set of all possible color combinations (i.e., color subsets) that can be used in any period. There are a total of 8 possibilities, including the combination that contains only the cleaned state. The binary parameter a_{cm} indicates whether color combination m includes c. Given the limited number of colors, determining the optimal color sequence for each combination and calculating the associated optimal cost ϕ_m is straightforward. The objective function incorporates additional costs related to bin utilization, item painting, and item storage. The unit cost of using bin b is denoted as μ_b, the unit storage cost for non-colored items of type j is denoted as $\hat{\psi} j$, and ψjc for colored items. The proposed MIP includes the following decision variables: Y_{bt}, binary variables equal to 1 if bin b is used in period t; Z_{jbt}, linear variables represent the quantity of items of type j cut from bin b in period t; X_{mt}, binary variables equal to 1 if model m is used in period t; Q_{jct}, linear variables indicating the quantity of colored items of type j in color c during period t; $\hat{Q}_{jt}$, linear variables indicating the quantity of non-colored items of type j during period t; I_{jct}, linear variables indicating the inventory of colored items of type j in color c at the end of period t; and $\hat{I}_{jt}$, linear variables indicating the inventory of non-colored items of type j at the end of period t. The integrated problem can be formulated as follows.

$$Min \sum_{b\in\mathcal{B}} \mu_b\, Y_b \;+ \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}} \phi_m\, X_{mt} \;+ \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \sum_{c\in\mathcal{C}} \psi_{jc} I_{jct} \;+ \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \hat{\psi}_j \hat{I}_{jt} \tag{1}$$

subject to :

$$\sum_{m\in\mathcal{M}} X_{mt} = 1 \qquad \forall t \in \mathcal{T} \tag{2}$$

$$\sum_{j\in\mathcal{J}} Q_{jct} \leqslant f_t \times a_{cm} \times X_{mt} \qquad \forall j \in \mathcal{J}, \forall c \in \mathcal{C}, \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \tag{3}$$

$$\sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}} \phi_m\, X_{mt} \;+ \sum_{j\in\mathcal{J}} \sum_{c\in\mathcal{C}} Q_{jct} \times p_j \;\leqslant\; f_t \qquad \forall t \in \mathcal{T} \tag{4}$$

$$I_{jc\ t-1} + Q_{jct} = d_{jct} + I_{jct} \qquad \forall c \in \mathcal{C}, \forall t \in \mathcal{T} \tag{5}$$

$$\hat{I}_{jt} + \sum_{c \in \mathcal{C}} Q_{jct} = \hat{Q}_{jt} + \hat{I}_{jt-1} \qquad \forall c \in \mathcal{C}, \forall t \in \mathcal{T} \tag{6}$$

$$\sum_{b \in \mathcal{B}} Z_{jbt} \geqslant \hat{Q}_{jt} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \tag{7}$$

$$w_j \times Z_{jbt} \leq g_{jb} \times Cap_b \qquad \forall b \in \mathcal{B},\ \forall t \in \mathcal{T},\ \forall j \in \mathcal{J} \tag{8}$$

$$\sum_{j \in \mathcal{J}} w_j Z_{jbt} \leq Cap_b \qquad \forall b \in \mathcal{B},\ \forall t \in \mathcal{T} \tag{9}$$

$$M\ Y_{bt} \geqslant Z_{jbt} \qquad \forall j \in \mathcal{J},\ \forall b \in \mathcal{B},\ \forall t \in \mathcal{T} \tag{10}$$

$$Y_b,\ X_{mt} \in \{0,\ 1\} \qquad \forall b \in \mathcal{B},\ \forall t \in \mathcal{T} \tag{11}$$

$$Q_{jct},\ \hat{I}_{jt},\ I_{jct},\ Z_{jbt} \in \mathbb{N} \qquad \forall j \in \mathcal{J},\ \forall c \in \mathcal{C},\ \forall t \in \mathcal{T} \tag{12}$$

The objective function (1) aims to minimize the total costs associated with bin utilization, color change, and storage of both colored and non-colored items. Constraints (2) ensure the selection of only one color combination per period. Constraints (3) allow painting items in color c during period t (i.e., $Q_{jct} > 0$) only if the selected color combination includes this color for that period (i.e., $a_{cm}.X_{mt} = 1$). The constraints (4)-(6) are related to lot sizing. Constraints (4) represent the period capacity constraints. Flux conservation for colored and non-colored items is modeled by constraints (5) and (6), respectively. Constraints (7) ensure that the buffer between the cutting phase and the painting phase is filled. Constraints (9) to (10) represent the cutting stock constraints. Constraints (8) enforce cutting items of type j exclusively from compatible bin b (i.e., $g_{jb} = 1$). Constraints (9) represent the bin capacity constraints. Constraints (10) set the binary variable Y_{bt} to one exclusively when bin b is used in period t (i.e., $X_{jbt} > 0$). Constraints (11)-(12) define the range of variables.

Computational experiments were carried out to assess the performance of the MIP using CPLEX 20.1.0 on an Intel Xeon 3.60 GHz processor with 128 GB RAM. The model, implemented in C++, was tested on 300 randomly generated instances based on a real-case study. These instances are categorized into 30 distinct classes, each comprising 10 instances. The classes are defined by the number of periods (each period corresponds to an 8-hour workday) and the number of item types: $|\mathcal{T}| = 5, 10, 15, 20, 25, 30$ and $|\mathcal{J}| = 10, 20, 30, 40, 50$. We observed that classes with equal numbers of periods but varying item quantities exhibited similar behavior, suggesting that the class complexity is primarily determined by the number of periods rather than the number of items. To present results concisely, instances with the same periods were grouped into a single row, as shown in Table 1. Each row in the table reports the mean computation times in CPU seconds (CPU) and the average deviation from the optimal objective values in percent (GAP). Additionally, we include the average number of constraints (NC) and the average number of variables (NV).

$\|\mathcal{J}\|$	$\|\mathcal{T}\|$	NV	NC	CPU	GAP
$[10, 50]$	5	18770	17775	11.72	0.00
$[10, 50]$	10	35964	36501	250.95	0.00
$[10, 50]$	15	53403	55106	1297.88	0.10
$[10, 50]$	20	71154	70804	1815.29	0.40
$[10, 50]$	25	87964	92890	2950.27	1.37
$[10, 50]$	30	104233	112522	3600.00	2.00

Table 1: Numerical results

3 Conclusion

Based on the experimental study below, the solver is unable to successfully solve the model within the given 1-hour time limit for instances with a size of 30 periods. To overcome this limitation, we have developed an approximate solution approach that effectively addresses the trade-off between the objective value and execution time. Our proposed method involves initially solving the lot sizing and sequencing part using branch and price method, which provides the necessary quantities of pieces to be cut in each period. This information is then utilized to solve the cutting stock problem, again employing the branch and price method.

References

[1] Gislaine Mara Melega and Silvio Alexandre de Araujo and Raf Jans, Classification and literature review of integrated lot-sizing and cutting stock problems, European Journal of Operational Research, 271, 1–19 (2018)

[2] Xinye Hao and Li Zheng and Na Li and Canrong Zhang, Integrated bin packing and lot-sizing problem considering the configuration-dependent bin packing process, 303, 581–592, European Journal of Operational Research (2022)

[3] Douglas Alem and Reinaldo Morabito, Risk-averse two-stage stochastic programs in furniture plants, 35, 773–806, OR Spectrum (2012)

[4] Matheus Vanzela and Gislaine Mara Melega and Socorro Rangel and Silvio Alexandre de Araujo, The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production, 79, 148–160, Computers & Operations Research (2017)

A 3-level integrated lot-sizing and cutting stock problem with supplier selection and distribution decisions

Silvio Alexandre de Araujo
UNESP - São Paulo State University, Mathematics Department, 15054-000, São José do Rio Preto, SP, Brazil and GERAD
silvio.araujo@unesp.br

Gislaine Mara Melega
HEC Montreal, Montreal, H3T 2A7, QC, Canada
gislainemelega@gmail.com

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

Reinaldo Morabito
UFSCar - Federal University of São Carlos, Production Engineering Department, 13565-905, SP, Brazil
morabito@ufscar.br

Abstract

In this paper, a generalized 3-level integrated lot-sizing and cutting stock problem (G3ILSCS) proposed in the literature is extended to take into account other relevant decisions of the supply chain. The extensions consist of the selection of the suppliers of the raw materials used in the cutting process and the distribution of the final products from the production plant to a warehouse. To solve the integrated problem, a hybrid heuristic is proposed aiming to overcome the difficulties present in the integrated problem, mainly comprising the high number of variables, the multi-level structure, and the integrality requirements. The hybrid method embeds two decomposition approaches in each iteration of the algorithm: the column generation and the relax-and-fix procedure. Due to the features of the problem, an innovative column generation procedure is present to manage the cutting patterns in the cutting stock problem, as well as the cargo configurations in the distribution problem. The models and solution approaches are analyzed in an extensive computational study aiming to evaluate the impact of incorporating other decisions of the supply chain into the integrated problem, as well as to assess the performance of the hybrid heuristic when seeking a solution to the integrated problem.

1 Introduction

The idea of integrating processes in a production plant is to take into account, simultaneously, the decisions related to the problems involved so as to capture the interdependency between the decisions in order to obtain a better global solution. The manufacturing setting addressed in this study has its production processes linked to the cutting of raw materials (objects) and the production planning of end products (final products). In these industries, objects of large sizes are kept in stock to be cut later into smaller pieces of different sizes, using cutting patterns, in order to meet internal demand. These pieces then go to downstream levels of the production plant in order to produce and assemble the final products. The production planning of final products takes into account the tradeoff between setup and inventory holding costs to meet the clients' demand, considering capacity limitations. Therefore, it is necessary to plan the acquisition, production, and cutting of these objects, as well as the production of final products, in order to minimize the negative effects of these processes, which can be seen as the waste of material, delays in downstream levels, high costs, among others. These two problems are known in the literature as the cutting stock problem and the lot-sizing problem, respectively ([6, 3, 4, 1]).

A generalized 3-level integrated problem, *G3ILSCS*, proposed in [5], is extended and computationally analyzed considering other relevant decisions/levels of the supply chain. The first extension consists of an alternative means to the acquisition of objects, besides producing these objects in the company, and comprehends the supplier selection of objects used in the cutting process. In this alternative level, suppliers can also provide the inputs (objects) to the production plant. The other extension addresses an additional level after the production of final products, related to the distribution of the final products from the production plant to a warehouse. The problem with all these features comprise an 4-level integrated lot-sizing and cutting stock problem with supplier selection and distribution (*G4ILSCS*).

2 Mathematical Models and Solution Methods

The *G3ILSCS* model consists of a production environment composed of three levels and multi-periods in a deterministic setting. Level 1 corresponds to the production planning of objects, that have to be produced considering a capacitated environment in order to fulfill the downstream level (level 2). Level 2 is associated with the cutting process, in which the produced objects are cut into pieces according to cutting patterns [2] by a cutting machine with limited resources. The cut pieces can be used as components to assemble the final products or directly as final products. It is at level 3 that the production of the final products occurs and the independent demand for final products has to be met in each time period. The link between the different time periods is provided by inventory at each level. There is a bill-of-material relationship, for which the dependent demand of final products triggers a dependent demand for pieces, and indirectly, for objects. Therefore, the decisions of the 3-level integrated problem determines simultaneously a production planning that defines for every time period: the production quantities for products and objects and the cutting patterns, with corresponding frequencies, considering limited resources, while searching for a global optimal minimal solution to the 3-level integrated problem.

In the *G4ILSCS*, an alternative means to the acquisition of objects needed in the cutting process is considered, i.e., objects can be produced at level 1 of the production plant, as well as be purchased from external suppliers at level 1A. The supplier selection at level 1A takes into account a set of suppliers offering different types of objects, which can be purchased considering fixed and variable costs, proportional to the quantity purchased. The price of objects does not vary according to the number of objects ordered, i.e., there is no discount rate. In this integrated approach, the supplier selection decisions define the number of objects to be purchased considering the different supplier costs, which allied to an optimized production planning of the cutting process, and the production of the final products and objects, aims to reduce the total costs in the integrated problem.

The other extension modeled in the *G4ILSCS* manages the distribution of final products. Level 4 is responsible for the distribution costs incurred from shipments between the production plant and the warehouse. The distribution decisions are related to the load/arrangements of final products into vehicles, i.e., they are associated with the number of vehicles needed to transport the final products, hence, such decisions are directly linked to the production lot-sizing decisions of final products. In this integrated problem, the demand of final products is shifted from the production plant to the warehouse and in both sites, there is the possibility of inventory. Inventory at the production plant arises from cargo consolidation on the vehicles, whereas at the warehouse, inventory is addressed to keep the final products in stock in order to meet the clients' orders. Therefore, the distribution decisions define the number of vehicles utilized in the transport of final products and the transportation costs incurred in each time period of the planning horizon, i.e., this problem combines the lot-sizing and cutting stock decisions with vehicle loading decisions.

Considering that the problems addressed in this study are classified as NP-hard, we proposed a hybrid heuristic solution method to solve the integrated problem, aiming to overcome the difficulties, comprising the multi-level structure, the hight number of the variables, and integrality requirements. The goal of the hybrid heuristic is to provide a good trade-off between solution quality and computational effort while solving the integrated problem. The column generation procedure is used as a first step to generate an initial matrix of columns (cutting patterns for cutting the objects into pieces and cargo configurations for loading the final products into de vehicles) at levels 2 and 4 of the integrated problem. After that, the column generation is applied in each step of the relax-and-fix procedure in the hybrid heuristic, aiming to find more attractive columns, for cutting patterns and cago configurations, while the hybrid heuristic searches for a feasible solution to the integrated problem.

The models and solution methods are analyzed in an extensive computational study. Therefore, the main objective of this study is to evaluate the impact of incorporating levels of the supply chain into the integrated problem, as well as assess the performance of the hybrid approach in different environments when solving the 3-level integrated lot-sizing and cutting stock problem with supplier selection and distribution decisions.

Acknowledgments

This research was funded by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (*CNPq*), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (*CAPES*) and the Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP process n 2016/01860-1, 2018/19893-9, 2019/03302-4, 2022/05803-3).

References

[1] N. Brahimi, N. Absi, S. Dauzère-Pérès, and A. Nordli. Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, 263(3):838 – 863, 2017.

[2] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849 – 859, 1961.

[3] A. M. Gomes, J. F. Gonçalves, R. Alvarez-Valdés, and J. M. V. de Carvalho. Preface to the special issue on cutting and packing. *International Transactions in Operational Research*, 23:3 – 4, 2016.

[4] R. Jans and Z. Degraeve. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46(6):1619–1643, 2008.

[5] G. M. Melega, S. A. de Araujo, and R. Jans. Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research*, 271(1):1 – 19, 2018.

[6] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109 – 1130, 2007.

Sustainability

Aula Maxima - Wednesday, 23/08 - 11:00-12:30

A Heuristic Algorithm to solve the One-Warehouse Multi-Retailer Problem with an Emission Constraint

Matthieu Gruson
ESG - UQÀM and CIRRELT
gruson.matthieu@uqam.ca

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

Ola Jabali
Politecnico di Milano and CIRRELT
ola.jabali@polimi.it

Quihua Zhong
HEC Montréal, CIRRELT and GERAD
quihua.zhong@hec.ca

Abstract

In this work, we consider the one-warehouse multi-retailer problem with a global carbon emission cap constraint (OWMR-EC). This constraint aims at limiting the carbon emissions related to the production, setup and inventory holding operations. We develop a penalized relaxation method to heuristically solve the considered problem, both with and without the possibility of having initial inventory. This heuristic uses in itself another heuristic that we propose to solve the standard one-warehouse multi-retailer problem (OWMR). Our penalized relaxation method is tested on numerous instances adapted from the literature. Our results indicate that the penalized method is able to find between 87.4 and 89.8 % of feasible solutions for this NP-hard problem, while achieving an average optimality gap of about 2%. Furthermore, the results indicate that the heuristic for the standard OWMR is also very effective. We also perform a sensitivity analysis on the optimal solutions of the OWMR-EC to better understand the implications of the carbon emission cap constraint. The sensitivity analysis indicates that the marginal cost of reducing carbon emissions increases as the emission cap decreases. The analysis also shows that the correlation between the cost and emission parameters has an important impact on the potential to further lower the emissions, compared to the emission of the minimum cost solution.

1 Introduction

Over the last decades, there has been a growing interest in incorporating sustainability issues in supply chain management. The main concerns relate to global warming and greenhouse gas (GHG) emissions which, if left at their current levels, will lead to climate changes. The wish to have a "greener" image has also led individual companies to reduce their carbon footprint and engage in more environmental friendly production processes.

Carbon emissions considerations have been introduced into the basic lot sizing models, primarily through constraints that limit the GHG emissions or through penalties in the objective function. The considered emissions relate to production (e.g., by using some machinery), setup (e.g., by using some extra power to set up a machine) and inventory holding (e.g., by using cooling or heating systems). One of the earliest work is the one of Benjaafar et al. [2] who study the impact of different policies on carbon emissions. The policies include the carbon cap policy (where the total carbon emission is limited by a fixed amount), the carbon tax policy (where there is a tax paid per unit of carbon emitted), the carbon cap-and-trade policy (where companies can emit more than the allowed cap but have to pay for it, or reversly firms that do not emit beyond the cap can sell their unused carbon units), and the carbon offset policy (where it is possible to buy carbon units from independent suppliers and/or invest into projects whose goal is to reduce carbon emissions). Later, Absi et al. [1] incorporate carbon emission constraints in the basic single item uncapacitated lot sizing problem (SI-ULSP) model. They impose emission limits globally, per period, and on a rolling horizon basis. In a similar way, Retel Helmrich et al. [7] address the SI-ULSP with an emission cap constraint imposed on the entire planning horizon.

Carbon emission considerations also appear in multi level lot sizing problems. The work of Memari et al. [6] is one such example. The purpose of this paper is to contribute to the green lot sizing literature by integrating a carbon emission constraint in the one-warehouse multi-retailer problem (OWMR). We call the resulting problem OWMR-EC. Our aim is to develop an efficient and easy to reproduce algorithm for this NP-hard problem.

2 A two-stage heuristic for the OWMR

We first propose a two-stage heuristic for the OWMR. In Section 3, we build upon this heuristic to devise a heuristic for the OWMR-EC. The objective of the two stages are to define a production plan for the warehouse, and to find a delivery plan for each retailer, respectively. Specifically, the output of the first stage is used as an input for the second stage.

The purpose of the first stage is to obtain a production plan for the warehouse. To simplify the complexity of the OWMR, we aggregate all retailers and treat them as one. We therefore define an aggregate setup cost, and an aggregate inventory holding cost. Treating all retailers as one, we obtain a two-level serial system which can be seen as an OWMR for which the number of retailers is equal to 1. We solve this OWMR using the multi-commodity formulation introduced by Cunha and Melo [3]. We call this first stage the single retailer aggregation stage (SRA).

In the second stage, we fix the production setup decisions for the warehouse obtained in stage one and proceed to make the delivery plan for each retailer. We first disaggregate the OWMR into $|R|$ independent subproblems, where R is the set of retailers. Indeed, when the production setup decisions are fixed and when there is no initial inventory available at the warehouse, the OWMR reduces to $|R|$ independent subproblems, denoted by $OWMR^r$. To solve each subproblem $OWMR^r$, we developed a time-partitioning relax-and-fix heuristic (TPRF) and we adapted a dynamic programming recursion for two-level uncapacitated problems.

The TPRF heuristic contains elements of the time-partitioning (TP) heuristic used by Federgruen and Tzur [4] for the OWMR and the relax-and-fix heuristic (RF) introduced by Stadtler [8] for a multi level lot sizing problem with a general product structure and several constrained resources. The time-partitioning heuristic decomposes the time horizon into smaller intervals. The original problem is solved on these smaller intervals and side constraints are added on the boundaries of these intervals to obtain a feasible solution. The relax-and-fix heuristic is an iterative approach that works with a limited number of binary setup variables. At each iteration of the relax-and-fix heuristic, some binary variables are set to a value obtained in previous iterations. The problem obtained is solved to optimality and an additional subset of binary variables are set to their current value for the next iterations. The process stops when there are no more free binary variables. To solve the $OWMR^r$ subproblems to optimality over each time interval, we list all the possible delivery plans for each retailer. We then evaluate the cost of each plan and chose the one with the lowest total cost. In our preliminary experiments, this method has given the best results in terms of computational time compared to the use of a general purpose solver to solve the $OWMR^r$ subproblems to optimality over each interval.

The second method we use to obtain the retailers replenishment plan is based on the dynamic programming recursion proposed by Melo and Wolsey [5] to exactly solve a two-level uncapacitated lot sizing problem. Specifically, we adapt the recursion to the $OWMR^r$ subproblems.

3 A heuristic for the OWMR-EC

We develop a penalized relaxation method (PR) to solve the OWMR-EC. We relax the emission constraint and penalize it in the objective function with a penalty factor

β ($0 \leq \beta \leq 1$). It results in a OWMR which is solved using the SRA-TPRF or SRA-DP heuristic.

In the PR heuristic, we iteratively solve a series of OWMR problems using our proposed heuristic. At each iteration, we check if the solution obtained satisfies the emission constraint and if so, we compute its associated cost. After M iterations, the feasible solution with the lowest cost is kept as the final solution of our heuristic. If, for all iterations, we fail to obtain a feasible solution for the original problem, then our heuristic fails to provide a feasible solution to the OWMR-EC. We further add diversification and intensification phases on the PR heuristic to obtain more feasible solutions.

References

[1] N. Absi, S. Dauzère-Pérès, S. Kedad-Sidhoum, B. Penz, and C. Rapine. Lot sizing with carbon emission constraints. *European Journal of Operational Research*, 227(1):55–61, 2013.

[2] S. Benjaafar, Y. Li, and M. Daskin. Carbon Footprint and the Management of Supply Chains: Insights From Simple Models. *IEEE Transactions on Automation Science and Engineering*, 10(1):99–116, 2013.

[3] J. O. Cunha and R. A. Melo. On reformulations for the one-warehouse multi-retailer problem. *Annals of Operations Research*, 238(1):99–122, 2016.

[4] A. Federgruen and M. Tzur. Time-Partitioning Heuristics: Application to One Warehouse, Multiitem, Multiretailer Lot-Sizing Problems. *Naval Research Logistics*, 46:463–486, 1999.

[5] R. A. Melo and L. A. Wolsey. Uncapacitated two-level lot sizing. *Operations Research Letters*, 38(4):241–245, 2010.

[6] A. Memari, A. R. Abdul Rahim, N. Absi, R. Ahmad, and A. Hassan. Carbon-capped Distribution Planning: A JIT Perspective. *Computers & Industrial Engineering*, 97:111–127, 2016.

[7] M. J. Retel Helmrich, R. Jans, W. van den Heuvel, and A. P. M. Wagelmans. The economic lot-sizing problem with an emission capacity constraint. *European Journal of Operational Research*, 241(1):50–62, 2015.

[8] H. Stadtler. Multilevel lot sizing with setup times and multiple constrained reosurces: internally rolling schedules with lot-sizing windows. *Operations Research*, 51(3):487–502, 2003.

Coordinating shipments in lot-sizing models

Wilco van den Heuvel
Econometric Institute, Erasmus University Rotterdam
wvandenheuvel@ese.eur.nl

Marcel Turkensteen
Department of Economics and Business Economics, Aarhus University
matu@econ.au.dkl

Rommert Dekker
Econometric Institute, Erasmus University Rotterdam
dekker@ese.eur.nl

Abstract

Inspired by sustainability goals, we consider the problem of coordinating shipments in a stochastic lot-sizing setting. There are multiple items, which are shipped periodically from a single supplier to satisfy customer demands. This demand is dynamic and stochastic, but we assume that demand distributions are known or can be estimated. Costs are associated with the amount of inventory of each item and with each order of an item. There is an opportunity to achieve environmental savings by combining orders implying fewer shipments. This leads to a bi-objective lot-sizing problem with coordinated shipments where both the amount of shipments as well as costs need to minimized, such that a service level constraint is satisfied. We study a static-dynamic version, where first the ordering periods are determined, and given these ordering periods the ordering plan per item should be obtained. The complexity of the problem lies in the fact that not each item may be ordered in a potential ordering period, as fixed ordering costs are incurred for each order placed. It turns out that our model can also be used to solve the coordinated uncapacitated lot-sizing problem, which is a joint replenishment problem (JRP) in a deterministic lot-sizing setting. We propose several heuristic approaches to solve the model based on dynamic programming and test the performance in a computational study, both on instances based on a practical case and on JRP instances from the literature.

1 Introduction

We consider the problem of determining when and how much to ship from suppliers to a warehouse. In our problem, there are multiple items (and one supplier for each of them), which are shipped periodically to a warehouse to satisfy the demand of customers. Demand of customers is dynamic and stochastic, but we assume that demand distributions are known for each item and each period. The cost are associated with the amount on inventory of each item and with each order of an item. However, based on a company case, there is also a second objective of minimizing the number of periods in which deliveries take place to achieve environmental savings.

In our version of the problem, the order periods for each item have to be set beforehand, but the quantity to be ordered can be determined when demand is known. So at the start of the planning period, we have to select the periods in which orders are placed. In the selected periods, we can then order the quantities necessary to bring our inventory amounts to predetermined order-up-to levels. This approach is known as the so-called static-dynamic one in the literature (see e.g. Tempelmeier and Horst [3]). To ensure that a sufficient share of demand is fulfilled, a fill rate is set. We assume that a shipment can be made instantaneously and the order quantity is unlimited.

In fact, the problem under consideration is a stochastic bi-objective version of the economic lot-sizing problem, as demand is stochastic and there are the objectives of cost and the number of deliveries. In summary, the input of the problem is as follows:

T: number of periods,

K: number of items (product categories),

d_{kt}: stochastic demand for item k in period t,

s_{kt}: set-up cost for item k in period t,

h_{kt}: inventory holding cost per unit for item k in period t.

2 The model

In order to model the problem, define c_{ij}^k as the cost of having a set-up at time i to cover demand of item k for periods $i, \ldots, j$ with the next set-up at time $j+1$. These are the cost s_{ki} of ordering item k at time i plus the expected costs of the ending inventories in periods $i, i+1, \ldots, j$, given that an order quantity is determined which is sufficient to satisfy a given percentage of demand between i and j.

Since we take a static-dynamic approach and the cost parameters c_{ij}^k can be computed (analytically or numerically) upfront for given distributions, we can model the problem as a MIP. In order to do that, we define the following decision variables:

x_{ijk}: binary variable equal to 1 iff the schedule for item k contains order cycle $[i, j]$

z_t: binary variable equal to 1 iff there is a shipment in period t.

We use an ϵ-constraint approach, meaning that we set a bound ϵ on the number of deliveries (i.e., $\sum_{i=1,\ldots,T} z_t \leq \epsilon$) and solve a cost minimization problem for each $\epsilon \in \{1, \ldots, T\}$ to get the Pareto frontier. For a given ϵ, the MIP model is as follows:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{T} \sum_{j=i}^{T} c_{ijk} x_{ijk} && && (1)\\
\text{s.t.} \quad & \sum_{j=1}^{T} x_{1jk} = 1 && \forall k && (2)\\
& \sum_{i=1}^{j} x_{ijk} = \sum_{i=j+1}^{T} x_{j+1,i,k} && \forall j, k && (3)\\
& \sum_{j=i,\ldots,T} x_{ijk} \leq z_i && \forall i, k && (4)\\
& \sum_{i=1,\ldots,T} z_i \leq \epsilon && && (5)\\
& x_{ijk}, z_i \in \{0, 1\} && \forall i, j, k && (6)
\end{aligned}$$

Similar models can be found in Tempelmeier and Horst [3], except that we have an additional constraint to bound the number of shipments.

Note that the efficient solutions on the Pareto frontier can also be used to solve the coordinated uncapacitated lot-sizing problem (CULSP), where a major setup cost is incurred for each shipment. Namely, one can show that an optimal solution of CULSP can be found among the efficient solutions. Hence, any solution approach for the above model also provides a solution approach for CULSP.

3 Heuristic solution approach

As larger instances cannot be solved in a reasonable amount of time by the MIP model, we propose a heuristic approach based on dynamic programming (DP). Note that there are two related problems that need to be solved. The overarching problem is to find a set of m shipment periods V in $\{1, \ldots, T\}$ which minimizes total cost. Given the ordering periods in V, for each item k we need to solve a subproblem in which the objective is to find the shortest path (note that not each ordering period has to be used for each item, as this will incur set-up cost).

In the DP approach we compute the optimal order schedule per item k during the course of the algorithm. To formally describe this approach, we introduce the following notation:

- v_j^m: minimum cost up till period j with at most m shipments,
- V_j^m: set of 'optimal' order periods in interval $[1, j]$ when having at most m order periods,

- $SP_k(V;j)$ optimal cost for item k in $[1,j]$ when only orders in periods of V are allowed.

Note that $SP_k(V;j)$ can be determined efficiently by a shortest path approach. The DP approach is summarized in Steps 1 and 2 below. When computing the cost v_j^m in (7) of Step 2, we take the 'optimal' order schedules V_{i-1}^{m-1} (i.e., a schedule with $m-1$ order periods for $[1,i-1]$), we add an order period in i, and compute for each item what the 'optimal' cost are in $[1,j]$ when using only order periods in $V_{i-1}^{m-1} \cup \{i\}$.

Step 1: Initialization for $j = 1, \ldots, T$
$v_j^1 = \sum_k c_{1jk}$
$V_j^1 = \{1\}$

Step 2: DP recursion
for $j = 2, \ldots, T$ and $m = 2, \ldots, j$ we have the recursion

$$v_j^m = \min \left\{ v_j^{m-1}, \min_{i=2,\ldots,j} \sum_k SP_k(V_{i-1}^{m-1} \cup \{i\}); j) \right\} \tag{7}$$

set $i^* = \arg\min_{i=2,\ldots,j} \sum_k SP_k(V_{i-1}^{m-1} \cup \{i\}); j)$
if $v_j^m < v_j^{m-1}$, then $V_j^m = V_{i^*-1}^{m-1} \cup \{i^*\}$, otherwise $V_j^m = V_j^{m-1}$

The DP approach can be further refined by not only keeping track of the best, but of the q best solutions. The total complexity of this refined heuristic is $O(qKT^5)$.

The algorithm has been tested both on instances based on a practical case and on CULSP instances from the literature (Boctor et al. [1] and Robinson et al. [2]). Preliminary computational tests show that the heuristic approach performs well with optimality gaps of not more than 1% when $q = 10$, and with further improvements being attained often leading to an optimal solution when we increase q to 20.

References

[1] Boctor, Laporte, and Renaud (2004). Models and algorithms for the dynamic-demand joint replenishment problem. International Journal of Production Research, 42(13):2667–2678.

[2] Robinson, Narayanan, and Gao (2007). Effective heuristics for the dynamic demand joint replenishment problem. Journal of the Operational Research Society, 58(6):808–815.

[3] Tempelmeier, Horst (2013). Stochastic lot sizing problems. In: Handbook of stochastic models and analysis of manufacturing system operations (pp. 313-344). Springer, New York, NY.

Network Design for Closed-Loop Supply Chain Network with Hybrid Retailers/Collection Centres

Mahdi Doostmohammadi
University of Strathclyde
m.doostmohammadi@strath.ac.uk

Mehdi Amiri-Aref
Kedge Business School
mehdi.amiri-aref@kedgebs.com

Abstract

Traditional supply chain system is the open-loop supply chain in which goods/commodities are shipped from manufacturers/suppliers (sourcing facilities) to distribution centres, then to retailers, and finally to end customers. However, sustainable supply chain management has received significant attention among both researchers and practitioners. This results in closing the loop by collecting used products from customers, transporting them to collection centres, and finally to sourcing facilities for recycling/remanufacturing/refurbishment. This research studies a closed-loop supply chain with hybrid retailers/collection centres, which embraces production planning and facility location problems, proposes a mathematical formulation for the problem, and then solves it using relax-and-fix and fixe-and-optimise heuristics. Preliminary computational results associated with a realistic case study are reported.

Energy

Aula Maxima - Wednesday, 23/08 - 14:00-15:30

Ensuring fair allocation of renewable energy in microgrids for supply planning

Natalia Jorquera-Bravo
Unité de Mathématiques Appliquées, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France.
CEDRIC, Conservatoire National des Arts et Métiers, 75003 Paris, France.
natalia.jorquera@ensta-paris.fr

Sourour Elloumi
Unité de Mathématiques Appliquées, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France.
CEDRIC, Conservatoire National des Arts et Métiers, 75003 Paris, France.
sourour.elloumi@ensta-paris.fr

Safia Kedad-Sidhoum
CEDRIC, Conservatoire National des Arts et Métiers, 75003 Paris, France.
safia.kedad_sidhoum@cnam.fr

Agnès Plateau
CEDRIC, Conservatoire National des Arts et Métiers, 75003 Paris, France.
agnes.plateau_alfandari@cnam.fr

Abstract

The transition to a new renewable energy model has brought forth new roles and opportunities for individuals, who can participate as users, producers or both, known as prosumers [5]. As a result, energy collectives have emerged to facilitate the production and self-consumption of renewable energy [1, 2, 3, 4]. Among the various types of small-scale energy collectives, microgrids have gained prominence. In these networks, multiple users share a distributed energy resource (DER) with the aim of establishing a semi-autonomous system that can operate both connected to the main power grid and in isolation. Numerous studies have focused on the design and management of these microgrids [7, 6, 8], primarily aiming to minimize costs for the community.

Since DERs offer more affordable energy, but may not be able to fully meet the demand of all users, they will therefore compete with each other for these cost-effective DER energy resources, particularly during peak demand hours.

Consequently, there is a need of considering fairness allocation among smart homes that share the DER in the microgrid. To the best of our knowledge, the closest study addressing this aspect is presented in [9], where fairness is measured by minimizing the discrepancy between the cost assigned to each home and the cost it would have incurred if it had been the only home in the microgrid.

In this work, we investigate a community consisting of different homes, a shared DER, and a common energy storage system. Each home has an energy demand to satisfy over a discrete planning horizon. The demand can be fulfilled either by using the DER, the battery, or by purchasing electricity from the main power grid. Excess energy can be stored in the battery or sold back to the main grid. The objective is to find a supply plan that provides a fair allocation of renewable energy while minimizing the total cost of the microgrid. We formulate the problem as a mixed-integer linear programming model, considering various fairness metrics such as the proportional allocation rule and the min-max fairness. We evaluate the obtained predictive models using real instances with up to 7 houses and a one-day time horizon with 15-minute time intervals. The data used for these instances are sourced from E4C[1] and pertains to a smart building located in France.

References

[1] Bernhard J. Kalkbrenner and Jutta Roosen, Citizens' willingness to participate in local renewable energy projects: The role of community and trust in Germany, Energy Research & Social Science, 13, 60-70 (2016)

[2] Rasmus Luthander and Joakim Widén and Joakim Munkhammar and David Lingfors, Self-consumption enhancement and peak shaving of residential photovoltaics using storage and curtailment, Energy, 112, 221-231 (2016)

[3] Mike B. Roberts and Anna Bruce and Iain MacGill, Impact of shared battery energy storage systems on photovoltaic self-consumption and electricity bills in apartment buildings, Applied Energy, 245, 78-95 (2019)

[4] Campos Inês and Pontes Luz Guilherme and Marín-González Esther and Gährs Swantje and Hall Stephen and Holstenkamp Lars, Regulatory challenges and opportunities for collective renewable energy prosumers in the EU,Energy Policy, 138, 111212 (2020)

[5] Anna Butenko, User-Centered Innovation and Regulatory Framework: Energy Prosumers' Market Access in EU Regulation, Tilburg Law and Economics Center (TILEC), 112, 221-231 (2016)

[1] Interdisciplinary center Energy4Climate of Institut Polytechnique de Paris and École des Ponts ParisTech

[6] A.D. Hawkes and M.A. Leach, Modelling high level system design and unit commitment for a microgrid, Applied Energy,86, 1253-1265(2009)

[7] Y. Zoka and A. Sugimoto and N. Yorino and K. Kawahara and J. Kubokawa, An economic evaluation for an autonomous independent network of distributed energy resources, Electric Power Systems Research, 77,831-838 (2007)

[8] Fausto Calderon-Obaldia, Jordi Badosa, Anne Migan-Dubois and Vincent Bourdin; A Two-Step Energy Management Method Guided by Day-Ahead Quantile Solar Forecasts: Cross-Impacts on Four Services for Smart-Buildings, Energies, 13 (2020)

[9] Di Zhang, Songsong Liu and Lazaros G. Papageorgiou; Fair cost distribution among smart homes with microgrid, Energy Conversion and Management, 80, 498-508(2014)

First results on energy-oriented lot-sizing and scheduling with energy storage

Stephan KÖPPEL
RPTU Kaiserslautern-Landau, Germany
stephan.koeppel@rptu.de

Florian SAHLING
RPTU Kaiserslautern-Landau, Germany
florian.sahling@rptu.de

Abstract

We propose a novel model formulation for a multiproduct, capacitated lot-sizing and scheduling problem that incorporates sustainable energy technologies and energy storage. The optimization problem accounts for renewable energy generation, limited energy storage, connection to the national grid, and energy trading. The objective is to determine a feasible production plan that minimizes holding, setup, and energy costs. A two-stage solution approach based on the fix-and-optimize heuristic (F&O) is applied. First numerical results are presented to demonstrate the performance of the proposed solution approach.

1 Introduction

Energy consumption and saving have become more important not only since the global energy crisis in 2022, but also in the view of current climate change and its consequences. The need to reduce energy consumption and to shift from fossil energy sources to sustainable energy technologies such as solar and wind power is widely discussed and promoted, and governments are introducing regulations such as emission taxes. Industry, in particular, has a large share and responsibility in the change process due to its enormous energy consumption. Electricity in particular plays a key role.

It is therefore in the interest of the manufacturing industry to improve and optimize its overall energy concept to contribute to a more sustainable energy management. Industrial companies can install their own on-site, decentralized power generation plants, e.g. based on photovoltaics (PV). To enable efficient integration and use of intermittent renewable energy sources, energy storage systems (ESS), especially based on lithium-ion batteries, play an essential role.

To bundle these aspects and to manage an energy-oriented production efficiently, a production planning is required which considers energy aspects and flows. In particular, in the area of energy-oriented simultaneous lot sizing and scheduling, there is a lack of literature that addresses the difficulty of coordinating the production schedule with the energy schedule (see [1]). A first model formulation that considers simultaneous lot sizing and scheduling, as well as the integration of renewables, energy trading, and storage, is proposed by [4]. However, their approach is not suitable for realistic scenarios due to its poor solution performance. Moreover, it hardly considers ESS specifications. Therefore, we present a new model formulation for a lot-sizing and scheduling problem that incorporates sustainable energy technologies and energy storage. Our first numerical results show that, in contrast to [4], the proposed solution approach is able to solve larger instances as well.

2 Problem statement and modelling

We present a model formulation for the so-called EO-CLSD-ESS (Energy-Oriented Capacitated Lot-Sizing with Sequence Dependent setup costs and energy-storage). The model formulation is based on the well-known CLSD proposed by [2]. The goal is to determine a production schedule that minimizes holding, setup, and energy costs and captures the energy flows shown in Figure 1.

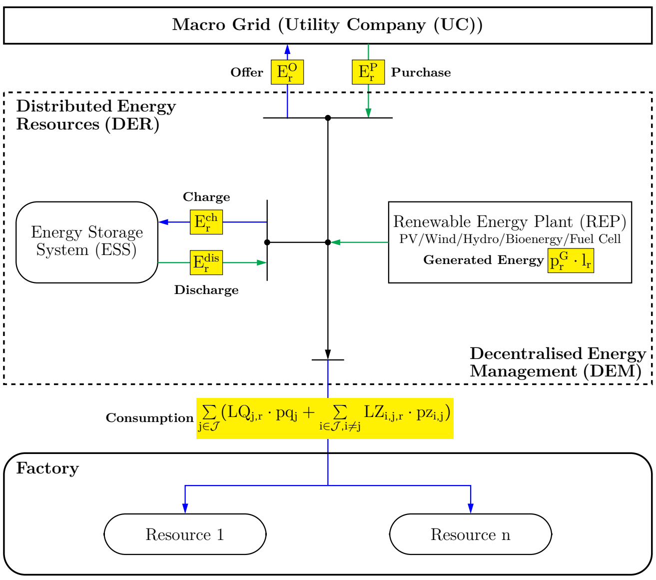


Figure 1: Energy flow. (cf. [4])

The main focus of EO-CLSD-ESS is to balance production and energy. Therefore, two types of planning periods are introduced: production-oriented macro periods t and (micro) energy periods r, each with constant power and length (60 min). Short energy periods r are necessary to better reflect (un)loading processes and energy prices with respect to volatile energy from sustainable energy technologies. Within a macroperiod t four different machine states are considered: "Production", "Setup", "Idle/Conservation", and "Off"; all four must be assigned to the corresponding energy period r in the correct sequence. Therefore, start and end points are defined to assign the energy supply from period r to the corresponding machine states. The figure 2 gives an overview.

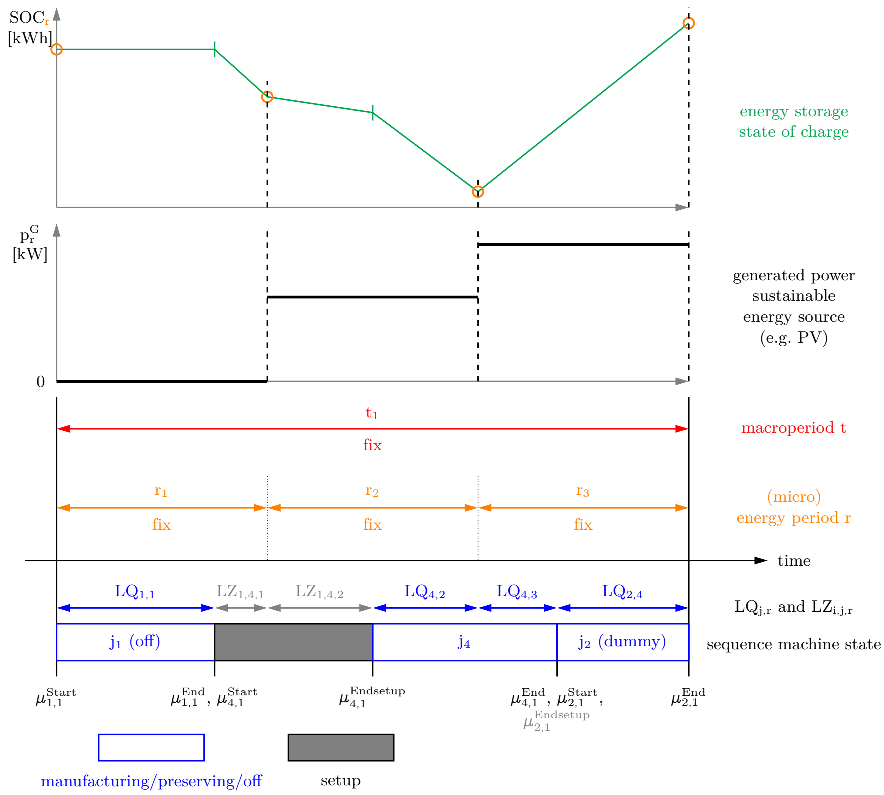


Figure 2: Period structure of EO-CLSD-ESS

3 Heuristic solution approach

Since it is impossible to solve the EO-CLSD-ESS to optimality even for smaller instances with a modern MIP solver in a reasonable time, a heuristic solution approach consisting of two steps is proposed. In the first step, energy aspects are not considered and only the basic CLSD model is solved using the MIP solver Gurobi. This solution

serves as an initial solution for the application of a Fix&Optimize heuristic (cf. [3]) with different decomposition strategies in the second step. Preliminary tests have shown that two variants are most suitable: a) period-oriented decomposition (Var. 1), b) product-oriented followed by period-oriented decomposition (Var. 2).
Three problem classes (PCs) are considered in our numerical study. The largest PC (PC III) consists of 20 products and 10 macro periods. This corresponds to a work week of 5 days with 2 shifts of 8 hours each. Within each PC, 64 randomly generated test instances (TIs) are examined. Preliminary computational results show that Gurobi, running on a high performance computing cluster at RPTU Kaiserslautern-Landau, could not find the optimal solution for a TI within a given time frame of 7200 seconds. Table 1 shows the high solution quality of our solution approach.

	PC I			PC II			PC III		
	Gurobi	Var. 1	Var. 2	Gurobi	Var. 1	Var. 2	Gurobi	Var. 1	Var. 2
Time^{Gur} [in sec]	7200	-	-	7200	-	-	7200	-	-
RelGap [in %]	3.43	-	-	12.14	-	-	7.43	-	-
Time^{Heu} [in sec]	-	123	226	-	725	1186	-	719	1772
$\overline{DevGH}^{UB}$ [in %]*	-	2.19	0.86	-	6.71	-0.27	-	1.57	-0.12

Table 1: Numerical results

*For each TI, the deviation of the objective function value obtained by the heuristic $objVal_{TI}^{Heu}$ from the Gurobi reference solution $objVal_{TI}^{Gur}$ is determined as follows: $DevGH_{TI}^{UB} = \frac{objVal_{TI}^{Heu} - objVal_{TI}^{Gur}}{objVal_{TI}^{Gur}}$.

Acknowledgement

The authors gratefully acknowledge the financial support from Carl-Zeiss-Stiftung within the project "Smarte Batchprozesse im Energiesytem der Zukunft" in the program "Durchbrüche".

References

[1] Bänsch, K., Busse, J., Meisel, F., Rieck, J., Scholz, S., Volling, T., Wichmann, M.G., 2021. Energy-aware decision support models in production environments: A systematic literature review. Computers & Industrial Engineering, 159, 107456.

[2] Haase, K., 1996. Capacitated lot-sizing with sequence dependent setup costs. OR Spectrum, 18(1), 51–59.

[3] Helber, S., Sahling, F., 2010. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. International Journal of Production Economics, 123(2), 246–256.

[4] Wichmann, M. G., Johannes, C., Spengler, T.S., 2019b. Energy-Oriented Lot-Sizing and Scheduling considering Energy Storages. International Journal of Production Economics, 216, 204–214.

Network Design for Closed-Loop Supply Chain Network with Hybrid Retailers/Collection Centres

Mahdi Doostmohammadi
University of Strathclyde
m.doostmohammadi@strath.ac.uk

Mehdi Amiri-Aref
Kedge Business School
mehdi.amiri-aref@kedgebs.com

Abstract

Traditional supply chain system is the open-loop supply chain in which goods/commodities are shipped from manufacturers/suppliers (sourcing facilities) to distribution centres, then to retailers, and finally to end customers. However, sustainable supply chain management has received significant attention among both researchers and practitioners. This results in closing the loop by collecting used products from customers, transporting them to collection centres, and finally to sourcing facilities for recycling/remanufacturing/refurbishment. This research studies a closed-loop supply chain with hybrid retailers/collection centres, which embraces production planning and facility location problems, proposes a mathematical formulation for the problem, and then solves it using relax-and-fix and fixe-and-optimise heuristics. Preliminary computational results associated with a realistic case study are reported.

Multi-level Manufacturing

WGB 1.07 - Thursday, 24/08 - 09:00-10:30

Integrated shelf-life rules for multi-level tablets manufacturing processes

Simonis, Michael
Department of Economics and Management, Karlsruhe Institute of Technology, Kaiserstraße 89, Building 05.20-4A 76133 Karlsruhe, Germany
michael.simonis@partner.kit.edu

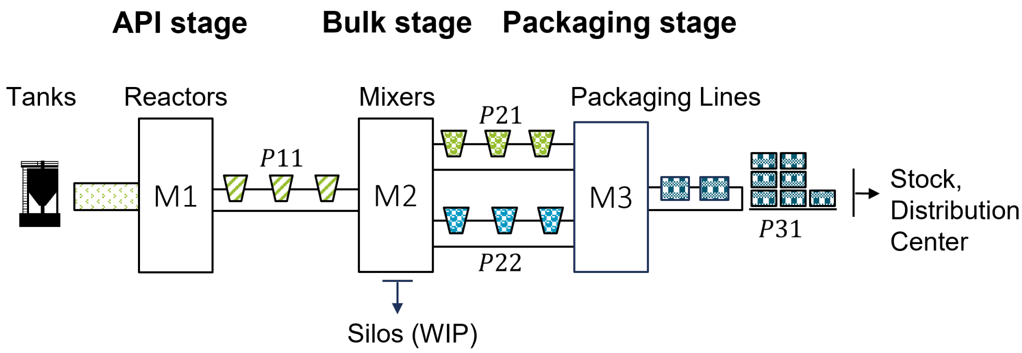
Nickel, Stefan
Department of Economics and Management, Karlsruhe Institute of Technology, Kaiserstraße 89, Building 05.20-4A 76133 Karlsruhe, Germany
stefan.nickel@kit.edu

Abstract

This paper discusses the multi-level capacitated lot-sizing problem with linked lot sizes and backorders (MLCLSP-L-B) considering deterministic product shelf-life applied on tablets manufacturing processes. Shelf-life is modelled by integrated shelf-life rules in tablets manufacturing processes. Thus, the MLCLSP-L-B is extended by integrated shelf-life rules (MLCLSP-L-B-SL). An exact mathematical problem formulation is provided. Three established benchmark approaches from literature, namely First In-First Out (FIFO), First-Expire First-Out (FEFO) and isolated shelf-life rules, are used to discuss model outcomes compared to MLCLSP-L-B-SL. Evaluation based on anonymized real-world data of five multi-level tablets manufacturing problem instances. Additionally, proposed solutions are compared in terms of manufacturing costs and shelf-life conflicts. Finally, planning rules and managerial insights are given for tablets manufacturing processes.

1 Introduction

Effective containment of globally raising prevalence of chronic symptoms and incidence of novel viral diseases requires products to remain stable in medicinal effects across varying treatment periods. Thus, governmental regularities around the world have defined that all prescription tablets have a shelf-life label to indicate when they are expire. Those regulatory authorities require comprehensive stable medicine for market approval. Moreover, [1] highlighted that tablets shortage situations caused by shelf-life issues became a huge image loss through publicity in the last decades.



(a) Example of a tablets manufacturing process

Products	Lay Time	Integrated Shelf-Life Rule	Shelf-Life	Remaining Shelf-Life (RSL)
$P11$	10	Constantly 40	40	$40 - 10 = 30$
$P21$	5	$10 + 0.5 \cdot \mathrm{RSL}(P11)$	$10 + 0.5 \cdot 30 = 25$	$25 - 5 = 20$
$P22$	20	$10 + 0.6 \cdot \mathrm{RSL}(P11)$	$10 + 0.6 \cdot 30 = 28$	$28 - 20 = 8$
$P31$	4	$5 + \min(\mathrm{RSL}(P21), \mathrm{RSL}(P22))$	$5 + \min(20, 8) = 13$	$13 - 4 = 9$

(b) Example of (remaining) shelf-life dependencies to all involved ingredients for one batch of $P31$

Figure 1: Illustration of shelf-life impacts on tablets manufacturing

Hence, pharmaceutical tablets manufacturers spend much effort steering manufacturing processes to avoid competitive disadvantage by delivering medicine with sufficient long shelf-life, see [2].

[3] grouped a tablets manufacturing process into three stages, namely the production of active pharmaceutical ingredients (API), the bulk, and the packaging stage. The stages consist of multiple machines, which can produce several products, but only one product at the same time. Figure 1a visualizes an example of such a manufacturing process: The API stage consists of reactors, which consume raw materials from tanks and produce two kinds of active ingredients through chemical reactions. Then, mixers produce two sorts of tablets by granulating, mixing, pressing, and enameling those ingredients. The tablets can either be stored in silos, or processed into finished goods in the packaging stage. The packaging stage consists of packaging lines, which put tablets and recipes either into folding boxes of different sizes. The finished goods can either be stored in stock or transported directly to distribution centers. Ingredients have a huge impact on finished good shelf-life stability in multi-level manufacturing processes, see [4]. These shelf-life dependencies on ingredients are modeled by *integrated shelf-life rules* in tablets manufacturing processes. If no dependencies are considered, the rule is named *isolated shelf-life rule*. Figure 1b illustrates

the general behavior of ingredient's impact on the remaining shelf-life of finished good $P31$: Whenever the API material $P11$ is produced, then isolated shelf-life rule returns fixed 40 periods. The material is stored for 10 periods. Hence the remaining shelf-life is 30 periods. This batch is consumed by bulk products $P21$ and $P22$. Now, the integrated shelf-life rule applies a formula on the remaining shelf-life of $P11$. The shelf-life equals 25 and 28, and the remaining shelf-life reduces to 20 and 8 periods due to storage time for $P21$ and $P22$, respectively. With an analog calculation logic, finished good $P11$ turns out to have a remaining shelf-life of 9 periods. If remaining shelf-life is greater than 0, then no *shelf-life conflicts* exist.

Planning teams often focus on usually one year to derive midterm tactical production plans. They identify production, stock, and backorder quantities for each material, machine, and period in the planning horizon, such that inventory, backorder, and setup costs are kept at a minimum, demands are fulfilled on time, capacities are not exceeded, and shelf-life issues are avoided. Nonetheless, the practice shows, that shelf-life conflicts occur once in a while due to a lack of modeling integrated shelf-life rules in MRP procedures in planning systems (classic MRP and MRP2 procedures apply only isolated shelf-life rules). If expired batches occur, then the proposed production plan is rejected by business due to very strict regulations, and significant efforts (R&S costs) have to be taken into account to reschedule the production system. Thus, planning teams elaborate production plans containing no shelf-life conflicts so that proposed lot sizes don't lack in cost-efficiency or even feasibility.

The MLCLSP-L-B is well established in literature and already used in a wide range of applications in process industries. It is a time-discrete model, that balances production quantities, inventories, backorders, and setup operations for each product and period of the planning horizon, such that setup, inventory, and backorder costs are kept at a minimum, deterministic demands are fulfilled, and resource capacities are not exceeded. Among this research, new solution approaches consider inventories affected by shelf-life and novel formulations were established, see [5] and [6]. This paper contributes to existing literature in three aspects. First, it provides the first exact model formulation of integrated shelf-life rules for the MLCLSP-L-B. Second, the paper discusses performance of common inventory policies, isolated, and integrated shelf-life rule model formulations. Third, it shares problem instances for lot-sizing from real-world tablets manufacturing processes and managerial insights derived from presented solution approaches based on these real-world problem instances.

2 Problem definition

This section provides the MILP formulation of the MLCLSP-L-B. Each material is allocated on exactly one machine, but one material might be issued by several successor materials. Table 1 summarizes model decision variables and parameters.

The model based on a direct extension of [7] and [8].It is formulated as follows:

$$\min Z = \min \left\{ \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} c_p^{su} x_{p,t}^{su} + c_p^{bo} x_{p,t}^{bo} + c_p^{inv} x_{p,t}^{inv} \right\}, \text{ s.th.} \tag{1}$$

$$x_{p,t-1}^{inv} + x_{p,t}^{bo} + x_{p,t-t_p^{as}}^{p} = x_{p,t}^{inv} + x_{p,t-1}^{bo} + d_{p,t} + \sum_{s \in \mathcal{P}_p^{suc}} r_{p,s} x_{s,t}^{p}, \tag{2}$$

$$\sum_{s \in \mathcal{P}_m} t_s^{su} x_{s,t}^{su} + t_s^{p} x_{s,t}^{p} \leq b_{m,t}, \tag{3}$$

$$x_{q,t}^{p} \leq M_{m,q,t} \left(x_{q,t}^{su} + x_{q,t-1}^{l} \right), \tag{4}$$

$$\sum_{s \in \mathcal{P}_m} x_{s,t}^{l} \leq 1, \tag{5}$$

$$x_{q,t}^{l} - x_{q,t}^{su} - x_{q,t-1}^{l} \leq 0, \tag{6}$$

$$x_{q,t}^{l} + x_{q,t-1}^{l} - x_{q,t}^{su} + x_{r,t}^{su} \leq 2, \tag{7}$$

$$x_{p,0}^{inv} = 0, x_{q,0}^{l} = \overline{x}_{m,q}^{l}, x_{p,0}^{bo} = 0, x_{p,T}^{bo} = 0, \tag{8}$$

$$x_{p,t}^{su} \in \{0,1\}, x_{p,t}^{l} \in \{0,1\}, x_{p,t}^{bo} \geq 0, x_{p,t}^{p} \geq 0, x_{p,t}^{inv} \geq 0,$$

$$\forall m \in \mathcal{M}, p \in \mathcal{P}, q, r \in \mathcal{P}_m, q \neq r, t \in \mathcal{T}.$$

(1) aims to minimize the sum of setup, inventory, and backorder costs. The material balance equation is covered by (2), capacity constraints are included by (3), (4) binds a positive production quantity to a setup in the same or a linked lot size in the last period, (5) satisfies that at most one linked lot size per period occur, (6) guarantees that a linked lot size is only allowed when a setup in the same period or a linked lot size in the last period take place and (7) synchronizes production runs that continue over more than two periods on a machine $m \in \mathcal{M}$. Moreover, (8) sets the initial inventory and setup state, and the initial and final backorder quantities, respectively.

References

[1] L. Colberg, L. Schmidt-Petersen, M. K. Hansen, B. S. Larsen, and S. Otnes. Incorrect storage of medicines and potential for cost savings. European Journal of Hospital Pharmacy, 24(3):167–169, 2017.

[2] S. Kopp. Stability testing of pharmaceutical products in a global environ- ment. RAJ Pharma, 5:291–294, 2006.

[3] C. J. Savage, K. J. Roberts, and X. Z. Wang. A holistic analysis of phar- maceutical manufacturing and distribution: are conventional supply chain techniques appropriate? Pharmaceutical Engineering, 26(4), 2006.

[4] N. Young and G. O'Sullivan. The influence of ingredients on product stability and shelf life. In Food and beverage stability and shelf life, pages 132–183. Elsevier, 2011.

[5] H. Tempelmeier and K. Copil. Capacitated lot sizing with parallel machines, sequence-dependent setups, and a common setup operator. OR spectrum, 38(4):828–829, 2016.

[6] S. Chen, R. Berretta, A. Mendes, and A. Clark. Integrating shelf life con- straints in capacitated lot sizing and scheduling for perishable products. In Data and Decision Sciences in Action 2, pages 33–46. Springer, 2021.

[7] D. Quadt, H. Kuhn, Capacitated lot-sizing with extensions: a review, 4OR 6 (1) (2008) 61–83.

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

$x^{su}_{p,t}$	Equals 1, if $p \in \mathcal{P}$ is prepared for setup in $t \in \mathcal{T}$, otherwise 0
$x^{l}_{p,t}$	Equals 1, if the production of $p \in \mathcal{P}$ is continued from t to $t+1$ on period domain $\mathcal{T}_0$, otherwise 0
$x^{p}_{p,t}$	Production quantity of product $p \in \mathcal{P}$ in period $t \in \mathcal{T}$
$x^{inv}_{p,t}$	Inventory quantity of a product $p \in \mathcal{P}$ in period $t \in \mathcal{T}_0$
$x^{bo}_{p,t}$	Backorder quantity of a product $p \in \mathcal{P}$ in period $t \in \mathcal{T}_0$
$\mathcal{M}$	Set of machines $\{1, \dots, M\}$
$\mathcal{P}$	Set of products $\{1, \dots, P\}$
$\mathcal{T}$	Set of periods $\{1, \dots, T\}$
$\mathcal{T}_0$	Set of periods including initial period $\{0, \dots, T\}$
$\mathcal{P}^{suc}_p$	Set of successors of a product $p \in \mathcal{P}$
$\mathcal{P}_m$	Set of products that can be produced on machine $m \in \mathcal{M}$
$b_{m,t}$	Capacity of machine $m \in \mathcal{M}$ in period $t \in \mathcal{T}$
$d_{p,t}$	Demand of product $p \in \mathcal{P}$ in period $t \in \mathcal{T}$
c^{su}_p	Setup cost for a product $p \in \mathcal{P}$
c^{inv}_p	Inventory holding cost for a product $p \in \mathcal{P}$
c^{bo}_p	Backorder cost for a product $p \in \mathcal{P}$
t^{su}_p	Setup time for a product $p \in \mathcal{P}$
t^{p}_p	Production time for a unit of product $p \in \mathcal{P}$
t^{as}_p	Advanced shift for a product $p \in \mathcal{P}$
$r_{p,q}$	Number of units of product $p \in \mathcal{P}$ required to produce one unit of successor product $q \in \mathcal{P}^{suc}_p$
$\overline{x}^{l}_{m,p}$	Initial setup for all $p \in \mathcal{P}_m$ on $m \in \mathcal{M}$, such that $\sum_{p \in \mathcal{P}_m} \overline{x}^{l}_{m,i} \leq 1$
$M_{m,p,t}$	Large number, e.g. $M_{m,p,t} = \min\{\sum_{\tau \in \mathcal{T}, \tau \geq t} d_{p,\tau}, b_{m,t}/t^{p}_p\}$ for $m \in \mathcal{M}$, $p \in \mathcal{P}_m$ and $t \in \mathcal{T}$

Table 1: Decision variables, sets and parameters of the MLCLSP-L-B

Network Design for Closed-Loop Supply Chain Network with Hybrid Retailers/Collection Centres

Andrea Balog
Confirm Centre for Smart Manufacturing, Ireland
a.balogh@cs.ucc.ie

Michele Garraffa
Confirm Centre for Smart Manufacturing, Ireland
michele.garraffa@cs.ucc.ie

Barry O'Sullivan
Insight SFI Research Centre for Data Analytics, Ireland
b.osullivan@cs.ucc.ie

Michele Garraffa
Dipartimento di Ingegneria Gestionale e della Produzione (DIGEP), Italy
fabio.salassa@polito.it

Abstract

We consider the No-idle Permutation Flowshop Scheduling Problem (NPFSP) with a total tardiness criterion. We present two Mixed Integer Linear Programming (MILP) formulations based on positional and precedence variables, respectively. We study six local search procedures that explore two different neighborhoods by exploiting the MILP formulations. Our computational experiments show that two of the proposed procedures strongly outperform the state-of-the-art metaheuristic. We update 63% of the best known solutions of the instances in Taillards' benchmark, and 77% if we exclude those instances for which we proved that the previous best known solutions are optimal.

A real-life batch-sizing and sequencing problem with buffer & sequence-dependent setup times

La Palombara N., Behiri W., Berraf-Belmokhtar S., Chu C., Sali M.
Université Gustave Eiffel, PlaniSense SAS
nicola.lapalombara@planisense.com

Abstract

The problem arises in the automotive industry and consists of organizing batch production on an injection machine to meet demand from the downstream assembly process with decoupling buffers between them. The considered time horizon is 24 hours, divided into 5 minutes time slots. Sequence-dependent setup times, minimum batch size, and stock coverage are considered. A multi-objective integer linear programming (ILP) model is proposed and then validated through experiments on various instance sizes.

1 Introduction & literature review

The problem arises in the automotive industry and concerns an injection machine that feeds the downstream assembly process with a decoupling buffer between them. The goal is to determine the start time and duration of each batch so that demand occurring every 5 minutes from the assembly is met, while buffers are kept within predefined ranges. One of the related problems in the literature is the lot-sizing and scheduling problem (LSSP) which aims to minimize production, setup, and holding costs by simultaneously optimizing lot sizes and their schedule [1]. In the discrete lot-sizing and scheduling problem (DLSP) the planning horizon is divided into small periods, only one setup per period is allowed, with the so-called all or nothing assumption [2]. Many studies have addressed the same problem with a different approach known as batch scheduling problem (BSP) which refers to schedule similar jobs contiguously in order to avoid setup times or setup costs [3]. The DLSP can be transformed into a BSP, as highlighted by [4]. Some lot-sizing and scheduling problems in the injection molding context implement similar characteristics, such as backorders, stock coverage, buffer, and batch constraints. These types of problems have been showed to be NP-hard [5]. Comparatively to BSP which considers due date jobs in input, we start from a given demand to simultaneously determine batch sizes and their sequence. The main difference from lot-sizing lies in the lengths and granularity of the time horizon. In addition, our work ignores stock holding costs, and backorders are allowed. Both setup state conservation and all-or-nothing assumption are considered. Finally, we implement a multi-objective optimization problem using the lexicographic method.

2 Model formulation & experiments

This model considers K physical products to be produced over a discrete time horizon $t \in [1, T]$. Setups are sequence-dependent and we introduce K fictitious dummy products which are used to indicate an idle state. Coherently, we define $j \in J | J = [1, K^2 + K]$ the set of all the possible states allowed on the resource and composed of $J^k = [1, K]$ the set of production states, $J^c = [K + 1, K^2]$ the set of setup states and $J^i = [K^2 + 1, K^2 + K]$ the set of idle states. We use the index k when j is in J^k. The demand for item k at time slot t is given by $d_{k,t}$. The buffer capacity, the stock coverage level, and the buffer level are defined by $BMax_k$, $BMin_{k,t}$ and $b_{k,t}$. The production rate q_k represents the number of products k that can be released in the buffer at each time slot. A binary matrix $F(J \times J)$ is introduced to define the rules of the machine states' transition. If $f_{j,j'} = 1$ then for two successive time slots the transition from the state j to the state j' is allowed, otherwise it is forbidden. The vector v_j contains respectively the information for minimum batch size, setup times and $v_j = 1$ for all the idle states. $I_{j,0}$ and i_0 represent the initial buffer levels and the initial state of the machine. At each time slot, a single state is activated by a binary variable $x_{j,t}$, which is 1 if state j is activated at time t, 0 otherwise. Finally, the variables $h_{k,t}$ and $w_{k,t}$ represent the backorders and coverage stockouts of product k at time t. Thus, the ILP model can be formulated as follows:

$$Min.Obj_1 = \sum_{j \in J^k} \sum_{t \in T} h_{j,t}$$

$$Min.Obj_2 = \sum_{j \in J^k} \sum_{t \in T} w_{j,t}$$

$$Max.Obj_3 = \sum_{j \in J^k} b_{j,T}$$

$$\sum_{t=1}^{v_j} x_{j,t} \geq v_j x_{j,1}, \qquad \forall j \in J \tag{1}$$

$$\sum_{t'=t}^{t+v_j-1} x_{j,t'} \geq v_j (x_{j,t} - x_{j,t-1}), \qquad \forall t \in \{2, \ldots, T - v_j\}, \quad \forall j \in J \tag{2}$$

$$\sum_{t'=t}^{T} x_{j,t'} \geq (T - t + 1)(x_{j,t} - x_{j,t-1}), \qquad \forall t \in \{T - v_j + 1, \ldots, T\}, \quad \forall j \in J \tag{3}$$

$$\sum_{j \in J} x_{j,t} = 1, \qquad \forall t \in \{1, \ldots, T\} \tag{4}$$

$$\sum_{j \in J} f_{i_0,j} x_{j,1} \geq 1 \tag{5}$$

$$\sum_{i \in J} f_{i,j} x_{i,t-1} \geq x_{j,t} - x_{j,t-1}, \qquad \forall t \in \{2, \dots, T\}, \quad \forall j \in J \tag{6}$$

$$b_{j,t} = I_{j,0} + \sum_{t'=1}^{t} q_j x_{j,t'} - \sum_{t'=1}^{t} d_{j,t'} + h_{j,t}, \qquad \forall j \in J^k \tag{7}$$

$$w_{j,t} \geq BMin_{j,t} - b_{j,t}, \qquad \forall t \in \{1, \dots, T\}, \quad \forall j \in J^k \tag{8}$$

$$b_{j,t} \leq BMax_j, \qquad \forall t \in \{1, \dots, T\}, \quad \forall j \in J^k \tag{9}$$

$$x_{j,t} \in \{0, 1\}, \qquad \forall t \in \{1, \dots, T\}, \quad \forall j \in J \tag{10}$$

$$h_{j,t}, w_{j,t}, b_{j,t} \in \mathbb{N}, \qquad \forall t \in \{1, \dots, T\}, \quad \forall j \in J^k \tag{11}$$

The three objective functions are prioritized with a lexicographic method and represent backorders minimization, coverage stockouts minimization, and maximization of buffers. Constraints (1), (2) and (3) ensure the minimum batch size and the respect of setup times. Constraints (4) ensure the activation of one and only one state at each time slot. Constraints (5) and (6) regulate the correct machine states transition. Constraints (7) define the flow conservation constraints for buffers. Constraints (8) define coverage stockouts and constraints (9)-(11) define the range of variables. To assess the proposed ILP's performances, 400 instances have been solved using Cplex as shown in Table 1.

Each class, composed of 50 instances is identified by a number of products and time slots. All the other parameters of the problem are integers randomly generated according to a uniform distribution inspired by real data: demands $\in [0, 4]$; setup times $\in [4, 8]$ time slots; minimum batch sizes $\in [10, 14]$; idle states $= 1$; production rates $\in [4, 7]$; buffer capacities $\in [150, 300]$; initial buffer levels $\in [0, 100]$. Stock

Class	Products	Time slots	Time(sec)	Gap(%)
1	3	72	2.04	0
2	5	72	10.64	0
3	7	72	35.27	0
4	5	90	55.61	0
5	6	90	211.34	0
6	5	108	436.40	0
7	6	108	993.26	0
8	[8,16]	288	3600	31.18

Table 1: Numerical results of the analyzed instances

coverage = 24 time slots. i_0, the machine state before starting the optimization $\in [1, K]$. Overall, calculation times increase as the combination of K and T increases. For the 72-time slot instances, the ILP model achieves optimal results at a maximum of 35 seconds, and the largest class optimally solved required on average more than 15 minutes. With class 8 we replicated the real-world data managed by the company on a daily basis. A time limit of 60 minutes was set for the solver and an average gap equal to 31.18% was obtained with respect to Obj_1. From this experimental results, we conclude the ILP model solves small and medium instances but it is not able to tackle real-life instances which suggests to develop heuristics approach to provide better solutions than those obtained with the solver using the proposed ILP.

3 Conclusions & future works

We propose a multi-objective ILP model to deal with a simultaneous batch sizing and sequencing problem where both setups and minimum batches extend over several consecutive time slots. We adopt a lexicographic optimization approach to prioritize objectives according to the company's requirements. Numerical experiments show the ability of the ILP to solve small and medium instances however we need to investigate a heuristics and/or metaheuristics approach to tackle real-life problems efficiently. Another immediate perspective is to propose a dynamic approach for rescheduling regarding demand changes with a rolling horizon. Next, we are going to extend our study to the parallel machine's environment while common cranes and operators have to be shared for setup operations.

References

[1] K. Copil, M. Worbelauer, H. Meyr, and H. Tempelmeier, "Simultaneous lotsizing and scheduling problems: a classification and review of models," *OR spectrum*, vol. 39, no. 1, pp. 1–64, 2017.

[2] B. Fleischmann. The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, 44(3):337–348, 1990.

[3] C. N. Potts and L. N. Van Wassenhove, "Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity," *Journal of the Operational Research Society*, vol. 43, no. 5, pp. 395–406, 1992.

[4] C. Jordan and A. Drexl. Discrete lotsizing and scheduling by batch sequencing. *Management Science*, 44(5):698–713, 1998.

[5] C. L. Monma and C. N. Potts, "On the complexity of scheduling with batch setup times," *Operations Research*, vol. 37, no. 5, pp. 798-804, 1989, INFORMS.

Inventory Control

WGB 1.07 - Thursday, 24/08 - 11:00-12:30

Inventory Analytics

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

Abstract

"Inventory Analytics" provides a comprehensive and accessible introduction to the theory and practice of inventory control. The book outlines the foundations of inventory systems and surveys prescriptive analytics models for deterministic inventory control. It further discusses predictive analytics techniques for demand forecasting in inventory control and also examines prescriptive analytics models for stochastic inventory control.

Inventory Analytics is the first book of its kind to adopt a Python-driven approach to illustrating theories and concepts via computational examples, with each model covered in the book accompanied by its Python code. A GitHub repository containing all Python code discussed complements the book. Originating as a collection of self-contained lectures, the book will be an indispensable resource for practitioners, researchers, teachers, and students alike.

The aim of this extended abstract is to showcase the content of this book.

1 Introduction

Inventory control[1] is a thriving research area that plays a pivotal role, as a building block, in supply chain planning. For this reason, it attracts the attention of both industry and academia.

Selected topics from inventory control are regularly covered in academic programmes, at both undergraduate and graduate levels, offered by business schools, industrial engineering, and applied mathematics departments.

Problems faced by managers who engage with the challenges posed by inventory systems are generally simple to state, but complex to address. Obtaining good solutions to these problems requires a blend of expertise drawn from a variety of quantitative disciplines, such as operations research, economics, mathematics, and statistics.

[1]These sections are excerpts from "Inventory Analytics" (`https://doi.org/10.11647/OBP.0252`) by R. Rossi, which is released under a Creative Commons Attribution 4.0 International (CC BY 4.0) license (`https://creativecommons.org/licenses/by/4.0`).

The majority of existing books in inventory control theory adopt, in my view, an overly mathematical and abstract style of presentation. This style appeals to researchers in the area, but makes these books often inaccessible to practitioners, as well as to some business school researchers who have not received advanced mathematical training such as that offered by applied mathematics, computer science, or industrial engineering curricula. A book with a more applied, hands-on focus is missing.

"Inventory Analytics" [1] aims to fill this void. It is aimed at those who want to learn the basics of modelling aspects of inventory control problems without needing to resort to the technical literature; at those who, despite lacking advanced mathematical training, want to access seminal findings in this field, and to apply well-established models by employing state-of-the-art solvers and modelling languages.

The book requires a working knowledge of Python; it is therefore aimed at readers who have, at the very least, taken a basic Python programming course. Apart from this, the book aims at stripping mathematical results to the bare minimum while preserving sufficient rigour, and at focusing on the practical relevance of these results in the context of the implementation of solution methods for problems typically faced by a manager who juggles with day-to-day inventory control challenges.

The book is structured as follows. It first provides a general introduction to inventory systems, followed by an overview of basic deterministic models. All these models are paired with their respective Python implementation, which can be tested on motivating examples that are presented throughout. The book is complemented by a GitHub repository that contains all Python code discussed [2]. After showcasing established models in deterministic inventory control, the reader is introduced to forecasting. Forecasting is often only briefly surveyed in existing books on inventory control; with the readers often directed to specialised textbooks, which are again often inaccessible to practitioners or individuals without suitable advanced mathematical training. However, forecasting is a crucial aspect of any practical inventory challenge. This work covers the most well-known forecasting models in a hands-on and visually appealing manner. The introduction of forecast errors paves the way to stochastic inventory control models, which are presented in the following sections. Once more, the most well-known stochastic inventory control policies are discussed in a hands-on fashion, with supporting code snippets and motivating examples. The last chapter briefly presents seminal results in the context of the control of multi-echelon inventory systems. Finally, an appendix provides the relevant formal backgrounds on a number of topics that are leveraged throughout the main chapters.

2 Inventory Analytics

This book originates as a collection of self-contained lectures. These lectures are divided into an introduction to inventory control, which outlines the foundations of inventory systems; followed by three chapters on deterministic inventory control, demand forecasting, and stochastic inventory control.

Beside Inventory, the title of the book refers to Analytics. This is nowadays a concept that has been inflated with a plethora of meanings, so that it becomes difficult to understand exactly what each of us means when we refer to it. The Cambridge Dictionary defines Analytics as "a process in which a computer examines information using mathematical methods in order to find useful patterns." However, this appears to be quite a restrictive definition for our purposes.

To better understand the nature of Analytics, it is useful to observe that Analytics is often broken down into three parts: descriptive, predictive, and prescriptive. Descriptive Analytics is concerned with answering the question: "what happened?" Predictive Analytics is concerned with answering the question: "what will happen?" Prescriptive Analytics is concerned with answering the question: "how can we make it happen?" These are clearly complex questions that cannot be answered by mere *number crunching* on a computer: to answer these questions a decision maker must leverage soft as well as hard skills.

Many tend to think that the Analytics phenomenon is a recent development related to widespread availability of computing power. However, in his work "De Inventione," the Roman philosopher Cicero states that "there are three parts to Prudence: Memory, Intelligence, and Foresight." It is clear that Memory is the skill required to answer the question "what happened?"; Foresight, that required to answer the question "what will happen?"; and Intelligence, that required to answer the question "how can we make it happen?" It appears then that Analytics is just a contemporary rebranding of an art that has been known for millenia. *Prudentia* is the ability to govern and discipline oneself by the use of reason. *Inventio* is the central canon of rhetoric, a method devoted to systematic search for arguments. Incidentally, *inventio* also means inventory. In fact, when a new argument is found, it is *invented*, in the sense of "added to the inventory" of arguments. *Prudentia* and *Inventio* are the foundations upon which the art of Rhetoric stands.

It must not surprise us then that Analytics plays a prominent role in inventory management. Inventory management finds its roots into the practice of late medieval and early Renaissance merchants. The invention of double-entry bookkeeping (alla Veneziana) is typically attributed to Frà Luca Pacioli (c. 1447 – 19 June 1517). Pacioli leveraged Johannes Gutenberg's new technology to disseminate and popularise accounting practices that had been in use among Venetian merchants for a long time. However, Pacioli did not simply disseminate existing practices, he reinterpreted these practices within the framework of Cicero's rethoric. In "De Inventione," Cicero explains that there are five canons, or tenets, of Rhetoric: *Inventio* (invention), *Dispositio* (arrangement), *Elocutio* (style), *Memoria* (memory), and *Pronuntiatio* (delivery). Pacioli's "Tractatus de computis et scripturis" (1494), is divided into two main sections: (i) the Inventory, and (ii) the Disposition — the influence of Cicero's work is apparent. Pacioli writes: "In order to conduct a business properly a person must: possess sufficient capital or credit, be a good accountant and bookkeeper, and possess a proper bookkeeping system." In "the Inventory," Pacioli writes "The merchant must prepare a list of his inventory. Items that are most valuable and easier to lose should be listed first. [...] The inventory should be carried out and completed in a single day. [...] The inventory is to include the day that the inventory was taken, the place, and the name of the owner." In contemporary terms, Pacioli describes a so-called "physical inventory," the process by which a business physically reviews its entire inventory — as opposed to so-called "cycle counts," which focus on specific subsets of items. In "the Disposition," Pacioli describes the necessary books and rules to implement double-entry bookkeeping.

Pacioli's work represents a quantum leap in the realm of *descriptive inventory analytics*, a discipline that would evolve into a fundamental part of inventory management. However, no progress was made in the realm of *predictive* and *prescriptive inventory analytics* until late 1800, when Edgeworth, in his "Mathematical Theory of Banking," used the central limit theorem to determine cash reserves needed to satisfy random withdrawals from depositors, thus embedding a *predictive* probabilistic model within a *prescriptive* mathematical model to support inventory control decisions.

From these early results, over the past 150 years, inventory control has evolved into an independent discipline. The aim of this book is to provide an introduction to this discipline.

After introducing the foundations of inventory systems, in chapter "Deterministic Inventory Control" we survey *prescriptive analytics* models for deterministic inventory control, in chapter "Demand Forecasting" we discuss *predictive analytics* techniques for demand forecasting in inventory control, which originate in the realm of time series analysis and forecasting. Finally, in chapters "Stochastic Inventory Control" and "Multi-echelon Inventory Systems" we survey *prescriptive analytics* models for stochastic inventory control.

References

[1] Roberto Rossi, Inventory Analytics, 184 pages. Open Book Publishers, Cambridge, UK (2021)

[2] Roberto Rossi, inventoryanalytics: a Python library dedicated to Inventory Analytics, `https://github.com/gwr3n/inventoryanalytics` (2022)

Stochastic Dynamic Programming Formulation for the (R, s, S) Policy Parameters Computation

Andrea Visentin
School of Computer Science & IT, University College Cork, Ireland
andrea.visentin@ucc.ie

Abstract

The (R, s, S) is a stochastic inventory control policy widely used by practitioners. In an inventory system managed according to this policy, the inventory is reviewed at instant R; if the inventory is lower than the reorder level s an order is placed. The order's quantity is set to raise the inventory level to the order-up-to-level S. This paper introduces a new stochastic dynamic program (SDP) algorithm to compute the (R, s, S) policy parameters for the non-stationary stochastic lot-sizing problem. In recent work, [1] present an approach to compute optimal policy parameters under such assumptions. We present the first formulation of the (R, s, S) problem as a functional equation of an SDP model. This model is an extension of Scarf's (s, S). A simple implementation of the model requires a prohibitive computational effort to compute the parameters. However, we can speed up the computations by using K-convexity property and memoisation techniques. The resulting algorithm is considerably faster than the state-of-the-art, extending its adoptability by practitioners.

1 Problem description

This work considers the single-item, single-stocking location, stochastic inventory control problem over a T-period planning horizon. The demand's stochasticity and non-stationarity of period t are modelled through the random variable d_t. Cumulative demand of periods t to the beginning of period j takes the form of $d_{t,j}$ with $j > t$. If the demand in a given period exceeds the on-hand inventory, the excess is backlogged and carried to the next period. Under these assumptions, the (R, s, S) policy takes the vectorial form form $(\mathbf{R}, \mathbf{s}, \mathbf{S})$, with $\mathbf{R} = (R_1, \ldots, R_T)$; where R_t , s_t and S_t denote respectively the length, the reorder-level and order-up-to-level associated with the t-th inventory review.

Policies are compared based on their expected cost. Stocktaking has a fixed cost of W. We denote by Q_t the quantity of the order placed in period t. Ordering costs are represented by a fixed value K and a linear cost, but we shall assume that the variable cost is zero without loss of generality. At the end of each period, a holding cost h is charged for every unit carried from one period to the next. In case of a stockout, a penalty cost b is charged for each item and period. We denote with I_t the closing inventory level for period t, making I_0 the initial inventory. The order quantity Q_t is fixed at every review moment before the demand realisation to raise the inventory level to S_t. The order is placed only if t is a review period and the open inventory is below the order level s_t.

We consider the problem of computing the optimal $(\mathbf{R}, \mathbf{s}, \mathbf{S})$ can be formulated as follow:

$$C_1(I_0) \triangleq \min_{(\mathbf{R},\mathbf{s},\mathbf{S})} f_1(I_0, Q_1, R_1) + E[C_{1+R_1}(I_0 + Q_1 - d_{1,1+R_1})] \tag{1}$$

Where $C_1(I_0)$ is the expected cost of the optimal policy parameters starting at period 1 with the initial inventory I_0. In general, $C_t(I_{t-1})$ represent the expected inventory cost of starting at period t with open inventory I_{t-1}. While, $f_t(I_{t-1}, Q_t, R_t)$ is the expected cost of a review cycle starting in period t and ending up in period $t+R_t$; it comprises review, ordering, holding and penalty cost for the review cycle. $C_t(I_{t-1})$ values can be computed recursively when all the policy parameters are computed using the following formula:

$$C_t(I_{t-1}) \triangleq f_t(I_{t-1}, Q_t, R_t) + E[C_{t+R_t}(I_{t-1} + Q_t - d_{t,t+R_t})]) \tag{2}$$

with $C_{T+1}(I_T) \triangleq 0$. For a given $(\mathbf{R}, \mathbf{s}, \mathbf{S})$ parameters set, this formulation allows to compute the expected policy cost. However, the number of combinations of parameters is exponential, making this approach unusable for the computation of optimal ones.

2 Heuristic technique

The heuristic introduced in this work aims to compute locally optimal R_t values to produce a near-optimal $(\mathbf{R}, \mathbf{s}, \mathbf{S})$ policy. The main idea is to move the assignment of the decision variable R_t at period t and do not fix all of them at the beginning of the time horizon. This can be done by transforming the recursive Equation 2 into:

$$\widehat{C}_t(I_{t-1}) = \min_{R_t} f_t(I_{t-1}, Q_t, R_t) + E[C_{t+R_t}(I_{t-1} + Q_t - d_{t,t+R_t})]) \tag{3}$$

Solving this recursion could lead to different optimal R_t for different opening inventory levels I_{t-1}.

Our heuristics consists of choosing a locally optimal R_t assuming that an order is placed in period t and the possibility of placing a negative order. We define these locally optimal replenishment cycles as R_t^a. Knowing the expected cost of future periods $\widehat{C}_j$ with $j > t$, it is possible to compute the optimal s_t and S_t for that specific replenishment cycle R_t using SDP. The best S_t is the value that minimizes $\widehat{C}_t(S_t)$, since we place an order to reach the point with the lowest future expected cost.

$$S_t = \arg\min_{I_{t-1}} \widehat{C}_t(I_{t-1}) \tag{4}$$

So, assuming that an order is placed, the best replenishment cycle is the one that has the lowest cost after the inventory level is topped up to S_t:

$$R_t^a \triangleq \arg\min_{R_t} \widehat{C}_t(S_t) \tag{5}$$

The computation of $\widehat{C}_t$ requires the expected costs of future periods $\widehat{C}_j$ with $j > t$, which are dependent on the optimal R_j. We relaxed the cost function by defining C_t^a as the expected cost of using local optimal R_j^a for all periods j after t. Given $C_{T+1}^a(I_T) = 0$, it is possible to compute the relaxed cost function in a backward way using the following approximate SDP functional equation:

$$C_t^a(I_{t-1}) \triangleq f_t(I_{t-1}, Q_t, R_t^a) + E[C_{t+R_t^a}^a(I_{t-1} + Q_t - d_{t,t+R_t^a})] \tag{6}$$

This formula computes a near-optimal replenishment schedule $\mathbf{R}^a$, and the set of order and order-up-to levels optimal for that given schedule. Due to the relaxation, $\mathbf{R}^a$ can differ from the optimal $\mathbf{R}$; however this event is rare.

The resulting approximate SDP formulation is more complex than the (s, S) one, making the computational effort required to solve it prohibitive. This is mainly due to the computation of the expected cycle cost; its computation involves three variables in each period: current inventory, order size and length of the replenishment cycle. This computational effort can be considerably reduced applying the K-convexity property. The deployment of search reduction and memoisation techniques further improve the performances, and it has a crucial impact on the applicability of this model.

3 Experimental Results

We aim to evaluate the policies computed by the heuristic and the computational effort required. We assess the computational effort required to compute a policy and under an increasing time horizon. We used the same testbed presented in [1].

For the experiments, we use as a comparison the branch-and-bound (BnB) technique presented in [1]. This is the only (R, s, S) solver for this problem configuration available in the literature. The solvers used are:**BnB-Guided** the branch-and-bound approach presented in [1], **SDP** the basic implementation of the SDP heuristic model, and **SDP-Opt**, the heuristic implementation deployed using the K-convexity property and the immediate cost memoisation.

Figure 1 shows the logarithm of the average computational time. The simple implementation of the heuristic can barely solve tiny instances before the time limit,

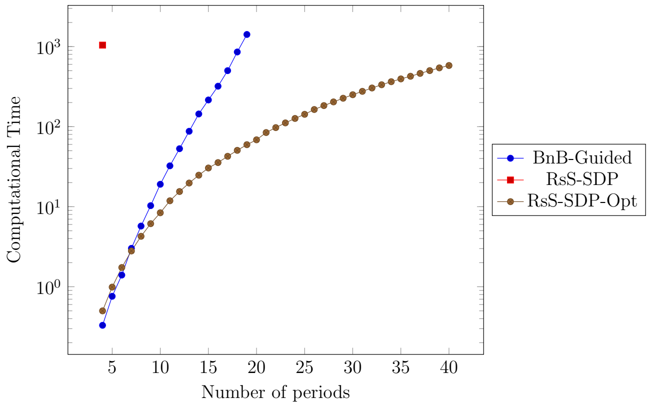


Figure 1: Computational time of the (R, s, S) SDP over the number of periods.

making it useless for every practical use. The reduction of computational effort provided by K-convexity and memoisation is massive. The guided BnB slightly outperforms the optimised SDP for small instances up to 8 periods, then the gap between the two strongly increases, making it able to solve instances more than twice as big in the same amount of time. The memoisation offers a great speed up in the computational times, which is more significant in bigger instances. For bigger instances, the physical memory needed grows to require the usage of memory swap and a slow down in performances.

In this testbed, the heuristic always computes the optimal replenishment plan.

4 Conclusions

This paper presented a heuristic for the non-stationary stochastic lot-sizing problem with ordering, review, holding and penalty cost, a well-known and widely used inventory control problem. Computing (R, s, S) policy parameters is computationally hard due to the three sets of parameters that must be jointly optimised. We presented the first pure SDP formulation for such a problem. The algorithm introduced solves to optimality a relaxation of the original problem, in which review cycles are considered independently, and items can be returned/discarded at no additional cost.

References

[1] Visentin, Andrea, Steven Prestwich, Roberto Rossi, and S. Armagan Tarim. Computing optimal (R, s, S) policy parameters by a hybrid of branch-and-bound and stochastic dynamic programming. European Journal of Operational Research 294, 91-99 (2021).

A Heuristic Method for Perishable Inventory Management Under Non-Stationary Demand

Suheyl Gulecyuz
Insight SFI Centre for Data Analytics, University College Cork.
suheyl.gulecyuz@insight-centre.org

Barry O'Sullivan
School of CS and IT, University College Cork.
b.osullivan@cs.ucc.ie

S. Armagan Tarim
Cork University Business School, University College Cork.
armagan.tarim@ucc.ie

Abstract

In our study we consider a finite-horizon periodic review perishable inventory system with non-stationary stochastic demand, zero lead time, FIFO issuing policy, and a fixed shelf life. We consider a fixed setup cost and ordering, holding, penalty, and outdating costs per item. We introduce a computationally-efficient heuristic for this problem which is based on the deterministic equivalent shortest path approach proposed by [1], the Wagner-Whitin algorithm and the Silver-Meal heuristic, in order to calculate the replenishment cycle and the order quantities for the (R_n, Q_n) policy that minimize the expected total cost throughout the planning horizon. We firstly determine the replenishment periods and cycles via a deterministic-equivalent approach which follows a novel procedure based on the Wagner-Whitin and Silver-Meal heuristics. Using the replenishment cycles determined in the first step, we then calculate the order quantities via a numerical search method and in an online fashion. Finally, we conduct numerical experiments for various scenarios and parameters. We conclude that the computation time is reduced significantly, and the average optimality gap between the expected total cost and the optimal cost is 1.87% which is the best result so far in the literature.

1 Introduction

We study a finite-horizon perishable inventory system with non-stationary stochastic demand with known parameters, penalty costs, the FIFO issuing policy, and a fixed shelf life. This simplifies the solution method and in general, it is not a relaxation with serious consequences. This problem can be solved to optimality by using the SDP approach, but it is not computationally feasible. Therefore, we employ a static-dynamic uncertainty strategy [2] and our objective is to find a near-optimal (R_t, Q_t) policy that minimizes the total cost throughout the planning horizon. We propose a computationally-efficient heuristic based on the Wagner-Whitin algorithm and Silver's heuristic [3]. The heuristic works in a recursive way and decides the near-optimal path up to period t with respect to the inventory level composition of each age at the end of period t that gives the least average cost per period as long as they satisfy the expected future demand. In our study, in order to deal with the large state space we assume the independence of replenishment cycles and hence it will allow us to focus on each cycle.

2 Problem Formulation

The stochastic model below formulates this problem:

$$\min E\{TC\} = \int_{d_T} \dots \int_{d_1} \sum_{t=1}^{T} \Bigg(q(Q_t) + h(I_{t,0})^+ + h \sum_{b=1}^{M-2} I_{t,b} - p(I_{t,1})^- + wI_{t,M-1} \Bigg) f_1(d_1) \dots f_T(d_T) dd_1 \dots dd_t$$

subject to

$$q(Q_t) = \begin{cases} a + cQ_t, & \text{if } Q_t > 0 \\ 0, & \text{otherwise} \end{cases} \qquad t = 1, 2, ..., T. \tag{1}$$

$$I_t = \sum_{b=0}^{M-1} I_{t,b} = \sum_{b=0}^{M-2} I_{t-1,b} + Q_t - d_t, \quad t = 1, 2, ..., T. \tag{2}$$

$$I_{t,0} = Q_t - (d_t - \sum_{b=0}^{M-2} I_{t-1,b})^+, \quad t = 1, 2, ..., T. \tag{3}$$

$$I_{t,b} = (I_{t-1,b-1} - (d_t - \sum_{j=b}^{M-2} I_{t-1,j})^+)^+, \quad t = 1, 2, ..., T; b = 1, ..., M-1. \tag{4}$$

$$I_{0,b} = 0, \quad b = 0, ..., M-1. \tag{5}$$

$$I_{t,b} \geq 0, \quad t = 1, 2, ..., T; b = 1, ..., M-1. \tag{6}$$

$$Q_t \geq 0, d_t \geq 0, \quad t = 1, 2, ..., T. \tag{7}$$

Note that $(x)^+ = \max(x, 0)$ and $(x)^- = \min(x, 0)$.

3 Calculating the Replenishment Cycle Order-Up-To Levels

We obtain the stochastic component of $c(i, j)$ and apply the multi-period Newsvendor formulation:

$$C_{ij}(S_{ij}) = \sum_{t=i}^{j} \left((h+p) \int_{-\infty}^{S_{ij} - \sum_{k=i}^{t} I_{M,k}} F_{i,t}(d_{i,t}) d(i,t) - p\left((S_{ij} - \sum_{k=i}^{t} I_{M,k} - d_{i,t}) \right) \right). \tag{8}$$

Then we obtain

$$\sum_{t=i}^{j} F_{i,t}\left(S^*_{ij} - \sum_{k=i}^{t} I_{M,k} \right) = \frac{Np}{h+p} \tag{9}$$

where the length of the replenishment cycle is denoted by $N = j - i + 1$. Since there is no explicit formula for the exact expected outdating [4], we use the EWA heuristic proposed by [5].

4 A Heuristic to Determine the Best Path and the Order Quantities

We denote the last period in which the expected demand is sufficed by the remaining inventory at the end of the replenishment cycle $R(i, t)$ by $T_{i,t}$, and the remaining inventory cost or the possible future replenishment cost given $\mathbf{I}_t$ at the end of period t by $c'(t+1, T_{i,t})$. In case there is not enough demand, $T_{i,t}$ represents the last period that is covered by the next replenishment cycle (i.e. $R(t+1, T_{i,t})$) selected via the modified Silver-Meal heuristic. Using the deterministic equivalent approach, we calculate each corresponding cycle costs $c(i, t)$ for all the possible replenishment cycles $R(i, t), i = max(1, t - M + 1), ..., t$. Using $\mathbf{I}_t$, we calculate $c'(t + 1, T_{i,t})$. We then calculate the average cost per period with respect to the formula below and using the total cost at the end of period $i - 1$, recursively:

$$AC(T_{i,t}) = \frac{TC(i-1) + c(i,t) + c'(t+1, T_{i,t})}{T_{i,t}}, \quad i \in \max(1, t-M+1), ..., t, \tag{10}$$

$$i^* = \arg\min_i AC(T_{i,t}). \tag{11}$$

and calculate the total cost as follows: :

$$TC(t) = TC(i^* - 1) + c(i^*, t), \tag{12}$$

and $TC(0) = 0$. We repeat this process for period $t + 1$ and continue until the end of the planning horizon.

Given the observed inventory composition at any period and using the pre-determined replenishment cycle plan found in Step 1 as an input, we apply the multi-period newsvendor formulation in order to find the order quantities.

5 Numerical Experiments

We compared our heuristic with the optimal SDP solution for 216 instances in total. The average optimality gap is 1.87%, no more than 5% at any instance, and the effect of the lifetime on calculation times are in terms of milliseconds in our heuristic unlike the SDP solution in which calculation times increase exponentially and quickly become unmanageable.

References

[1] Rossi R, Tarim SA, Hnich B, Prestwich S (2011) A state space augmentation algorithm for the replenishment cycle inventory policy. International Journal of Production Economics 133(1):377–384.

[2] Bookbinder JH, Tan JY (1988) Strategies for the probabilistic lot-sizing problem with service-level constraints. Management Science 34(9):1096–1108

[3] Silver E (1978) Inventory control under a probabilistic time-varying, demand pattern. Aiie Transactions 10(4):371– 379.

[4] Nahmias S (1975) Optimal ordering policies for perishable inventory—ii. Operations research 23(4):735–749.

[5] Broekmeulen RA, Van Donselaar KH (2009) A heuristic to manage perishable inventory with batch ordering, positive lead-times, and time-varying demand. Computers Operations Research 36(11):3013–3018.

Machine Learning

WGB 1.07 - Thursday, 24/08 - 14:30-15:30

Learning capacity consumption in multi-level capacitated lot-sizing problems

Tremblet, David
IMT Atlantique, LS2N-CNRS, La Chantrerie, 4 Rue Alfred Kastler, B.P. 20722, 44307 Nantes, France
david.tremblet@imt-atlantique.fr

Dolgui, Alexandre
IMT Atlantique, LS2N-CNRS, La Chantrerie, 4 Rue Alfred Kastler, B.P. 20722, 44307 Nantes, France
alexandre.dolgui@imt-atlantique.fr

Thevenin, Simon
IMT Atlantique, LS2N-CNRS, La Chantrerie, 4 Rue Alfred Kastler, B.P. 20722, 44307 Nantes, France
simon.thevenin@imt-atlantique.fr

Abstract

Capacitated multi-level lot-sizing problems determine the production quantities to minimize inventory and backlog in a multi-echelon bill of materials. The models typically require the production plan to respect resource capacity. The resulting production plan is then forwarded to the scheduling level, where a scheduler assigns a machine and a start date to each operation. Often, the scheduler cannot adhere to the plan, because the standard models for multi-level capacitated lot-sizing approximate the capacity consumption roughly. For instance, these models do not consider the precedence relations in the bill of material. This study investigates supervised learning approaches to predict the capacity consumption of multi-level lot-sizing problems. We fit machine learning models to predict the capacity consumption of plans for one period based on lot sizes as input data. To incorporate the fitted model into the MILP, we translate it into a set of constraints and variables. These additional constraints replace the classical capacity constraints in the MILP model for the multi-level lot-sizing. Our experimental results show that this integrated approach results in plans that are more reliable and cost savings.

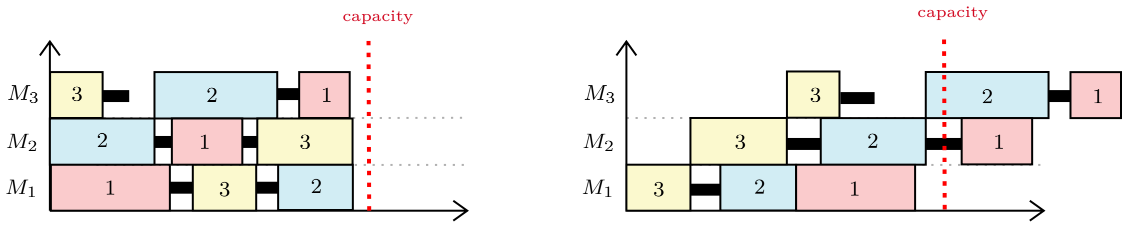
1 Introduction

Production planning determines the optimal quantities of items to produce in each period of the planning horizon. To guarantee the feasibility of the plans, capacity constraints limit the quantities to produce on the resources. Standard mathematical formulations for lot-sizing models consider a simple aggregation of the capacity consumption per resource. This simple constraint is sufficient when there is stationary bottlenecks in the production stages. However, such a constraint does not account for detailed scheduling constraints such as precedence relations. As a result, there is no guarantee that the production plans generated at the upper level respect these scheduling constraints, and this can lead to infeasible plans. Several studies consider the mathematical models for the integration of production planning and scheduling ([1], [6]), but these models consider complex formulations that are only relevant for small-size instances. In addition, these models lack scalability since they are only applicable in the case of standard scheduling problems, such as job-shop or flow-shop. In this study, we propose to incorporate a more precise capacity constraint learned from scheduling examples with machine learning models. More precisely, we propose a new data-driven formulation for the multi-level lot-sizing problem which integrates a machine learning model translated into a set of constraints and variables previously trained to predict the makespan of the scheduling problem. This new approach accurately predicts the capacity consumption for each period, and it results in production plans that are more often feasible and have lower costs that the plans obtained with the standard models from the literature.

2 Motivation

In this study, we extend the Multi-Level Capacitated lot sizing problem (MLCLSP) initially proposed by Billington1983[3]. This MLCLSP aims at determining the optimal production lot sizes, taking into account inventory holding costs, fixed setup costs, and unit production costs while satisfying the customers' demand. The production plan is define for given planning horizon, denoted by T, divided into smaller time periods. Multi-level lot sizing problems consider a bill of material that describes the components required to produce each end item. In each period, the production quantities must respect capacity limit for each resource. This capacity constraints ensure the feasibility of the plan, and a plan is feasible if it can be translated into a detailed schedule that can produce all lot within a time period. However, the standard capacity constraints the MLCLSP simply sums up the processing times required by each item for each resource, and it ignores the precedence relationship given by the bill of material (or bill of processes). Therefore, the lot sizes determined by these models usually exceed the capacity, leading to late demand satisfaction and unreliable production plans.

Figure 1 shows an example of a solution returned by the classical mathematical model for MLCLSP with 3 machines and 3 end items. In this example, the bill of process requires three production steps in a serial structure. Subfigure 1a illustrates the quantities of items determined for one period using the standard capacity constraints of the MLCLSP. This plan is optimal with respect to the MLCLSP constraints since it doesn't consider scheduling level characteristics such as precedence constraints or resource requirements. However, when scheduling the lot on the machines, the resulting production plan shown in Subfigure 1b violates the capacity of the period. Such infeasible plans are undesirable since they require manual modification of the plan, which is complex and often lead to costly plans.



(a) Solution of the MLCLSP

(b) Solution after scheduling the lots

Figure 1: Example of an infeasible solution of MLCLSP after forwarding lots to the scheduling problem

3 Integrated Machine learning and Lot-sizing

To provide feasible production plans, we aim to build an accurate estimation of the capacity consumption, and to integrated the resulting function in the MLCLSP model. We train machine learning models on datasets that correspond to instances of scheduling problem and the associated makespan. The model is trained to predict the makespan. Given relevant input features such as lot sizes or available inventory for each product, the machine learning model provides an accurate forecast of the value of the makespan. This makespan corresponds to the capacity consumption of each period of the production plan since it indicates if all production lots can be produced in a period. We translate the fitted model into a set of linear equations and variables, and we add them to the mathematical model for the MLCLSP to replace the standard capacity constraints ([4],[2]). We investigate the use of both linear regression and neural networks to predict the makespan.

Machine learning models are usually trained to minimize the mean squared error between the data sample output and the prediction. However, the resulting model is prone to underestimate the capacity consumption of the data sample, leading to lot sizes for products that can exceed the capacity. To ensure that our models always overpredict the capacity consumption on the training dataset, we build our machine learning models to minimize an error that drastically penalizes predicted values that underestimate the desired output. This loss function is related to a tilted absolute value function with a sufficiently high quantile [5]. In addition, to efficiently train our model, we propose a procedure that iteratively solves lot-sizing problems in order to generate adversarial examples that are underpredicted by our approach. These examples are used to complete the training dataset, leading to robust machine learning models and a data-driven lot sizing approach able to find production plans with a high percentage of feasibility while decreasing the total costs compared to the classical approach from the literature.

Acknowledgements

The present work was conducted within the project ASSISTANT (`https://assistant-project.eu/`) funded by the European Commission, under grant agreement number 101000165, H2020 – ICT-38-2020, Artificial intelligence for manufacturing. The authors would also like to thank the region Pays de la Loire for their financial support.

References

[1] Christian Almeder et al. "Lead time considerations for the multi-level capacitated lot-sizing problem". In: *European Journal of Operational Research* 241.3 (Mar. 2015), pp. 727–738.

[2] Max Biggs, Rim Hariss, and Georgia Perakis. "Constrained optimization of objective functions determined from random forests". In: *Production and Operations Management* (Oct. 2022).

[3] Peter J. Billington, John O. McClain, and L. Joseph Thomas. "Mathematical Programming Approaches to Capacity-Constrained MRP Systems: Review, Formulation and Problem Reduction". In: *Management Science* 29.10 (Oct. 1983), pp. 1126–1141.

[4] Matteo Fischetti and Jason Jo. "Deep neural networks and mixed integer linear optimization". In: *Constraints* 23.3 (2018), pp. 296–309. ISSN: 1383-7133.

[5] Roger Koenker and Kevin F Hallock. "Quantile Regression". In: *Journal of Economic Perspectives* 15.4 (Nov. 2001), pp. 143–156.

[6] Edwin David Gómez Urrutia, Riad Aggoune, and Stéphane Dauzère-Pérès. "Solving the integrated lot-sizing and job-shop scheduling problem". In: *International Journal of Production Research* 52.17 (2014), pp. 5236–5254.

A machine learning approach for identifying the best solution heuristic for a large scaled Capacitated Lotsizing Problem

Jens Kärcher
University of Hohenheim
jens.kaercher@uni-hohenheim.de

Herbert Meyr (Presenter)
University of Hohenheim
h.meyr@uni-hohenheim.de

Abstract

For some NP-hard lotsizing problems, many different solution heuristics exist, but they have different solution qualities and computation times depending on the characteristics of the problem instance. The computation times of the individual solution heuristics increase significantly with the problem size, so that testing all available solution heuristics for very large problem instances requires extensive time. Therefore, it is necessary to develop a method that allows a prediction of the best solution heuristic for the respective problem instance without testing all available solution heuristics. The Capacitated Lotsizing Problem (CLSP) is chosen as the problem to be solved, since it is well researched and several different solution heuristics exist for it. Five different forecasting methods are presented. One of them is a two-layer neural network called CLSP-Net. It is trained on small problem instances, which can be solved very fast with the considered solution heuristics. Nevertheless, CLSP-Net is able to identify the best solution heuristic even for very large problem instances.

1 Introduction

In the literature on lotsizing there are a large number of optimization problems which can be classified as NP-hard and for which many different solution heuristics have already been developed (see, e.g., [11, 9, 10, 1, 3]).

A promising solution approach for simultaneous lotsizing and scheduling models, which performs better in terms of solution quality and runtime compared to previous heuristics, is presented in [15]. Besides decomposing a multi-line problem[1] into

[1]The authors consider the General Lotsizing and Scheduling Problem for Parallel production Lines (GLSPPL).

independent single-line problems, the authors aggregate the time grid of the original model and reduce the number of periods in this way. Then, the aggregated problem is solved and the resulting production schedule is used to define line-dependent demands for all products. Since this heuristic leads to long computation times for a large number of products, [16] applies a product aggregation instead of a time aggregation, in which products are grouped into so-called setup families. This aggregation reduces the amount of data, but at the same time problems and errors can occur during the subsequent transfer of the planning results to the individual products (see [16, chapter 4]).

Therefore, the fundamental question arises, according to which aggregation rules the initial data should be aggregated and with which procedures the results should be disaggregated, in order to keep the aggregation error as low as possible on the one hand and to obtain a solution quality as high as possible on the other hand. The problem is that there are many different aggregation and disaggregation possibilities, which depend on the problem under consideration, and therefore it is not clear which of the rules or procedures should be applied step by step for a specific problem.

For this reason, it would be advantageous if a method exists that determines or predicts the best combination of aggregation and disaggregation rules depending on the characteristics of the considered problem. It is to be investigated whether machine learning can be used to achieve this goal. Since the described use case of decomposition in combination with aggregation and disaggregation is very complex and intransparent, it makes sense to first develop and test the desired method using a simpler problem environment. Then, the knowledge and experience gained can be used to develop a forecasting approach for the decomposition use case.

A very similar problem occurs when for an NP-hard problem, depending on the characteristics of a problem instance, different solution heuristics have the best solution quality. In this case, it is also unobvious which of the available solution heuristics should be applied to the considered problem instance.

As shown in Figure 1, the computation times of the solution heuristics (1-4) increase strongly with the problem size, which is why a computation of all available heuristics would require a huge amount of time, especially for larger problem instances. In application areas where an optimization task has to be solved very fast (e.g. real-time-scheduling), usually not all heuristics can be tested. Moreover, if even exact solution methods (in Figure 1: Gurobi) are not able to solve larger problem instances in a given time limit, a prediction of the best solution heuristic for the considered problem instance would be very helpful. This problem is referred to in the literature as Algorithm Selection Problem [12], Automated Algorithm Selection [13, 4] or Adaptive Recommendation Model [2].

The question is how to learn which type of problem instance works best with which solution heuristic. One approach could be to use a data collection that consists

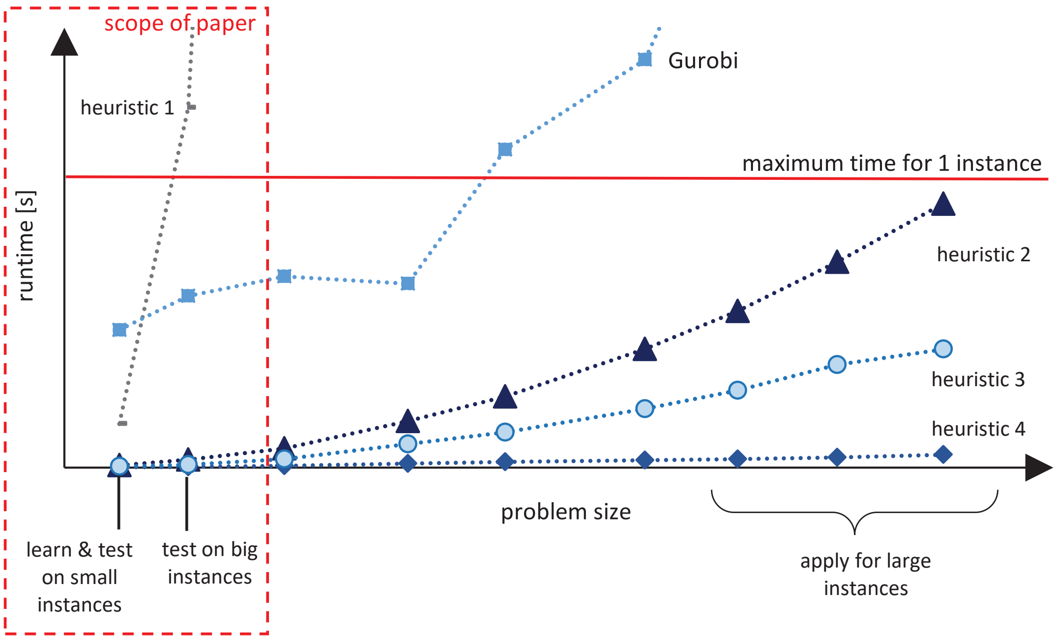


Figure 1: Basic idea.

of small problem instances, but still has different problem characteristics. For this training set, all solution heuristics can be tested quickly due to the short computation times, so that it is known which problem instance works best with which heuristic (input-output pairs). If patterns can be derived from the input-output pairs in the next step, these insights can be used to forecast the best heuristic for a very large problem instance for which no best solution heuristic is yet known. This approach is called supervised learning in the field of machine learning [7, p.105 f.]. To be able to check whether the forecasting method to be developed can also be applied to very large problem instances, it must be tested using problem instances that are significantly larger than the small instances of the training set (see Figure 1).

The Capacitated Lotsizing Problem (CLSP) is used as the problem to be solved. A major advantage of the CLSP is that it has important characteristics of a lot-sizing problem, but can be classified as low in terms of complexity. Furthermore, the CLSP is already very well researched, which is why many solution heuristics exist for solving the CLSP, some of which have very short computation times for small problem instances.

2 Summary and Conclusions

Five different prediction methods are presented for selecting the best solution heuristic depending on the characteristics of the considered problem instance for the CLSP. For training and validation of the forecasting methods, an extensive data collection (J72T12G24) is created, which considers the lotsizing of fictitious tire manufacturers. The J72T12G24 data collection considers six different demand scenarios, two scenarios regarding utilization, and three different scenarios regarding the ratio of setup and holding costs. The data collection includes a total of 7,200 problem instances, which have very different characteristics due to the combination of scenarios. All 7,200 problem instances are solved using the heuristics of [14], [5], [6] and [8]. The results show that the solution heuristics have different solution qualities depending on the characteristics of each problem instance, and none of the heuristics dominates the rest of the heuristics.

The best recognized approach to select the best solution heuristic depending on the characteristics of the considered problem instance is based on a two-layer neural network (CLSP-Net). As input, CLSP-Net is given 17 key performance indicators (KPIs), which are to be computed beforehand for each problem instance. The use of KPIs ensures that CLSP-Net can be used for different problem sizes, since the number of KPIs to be determined is independent of the problem size and therefore no adjustments to the network structure of CLSP-Net are necessary. As an output, CLSP-Net determines a probability for each of the four heuristics, whereby the heuristic with the highest value has to be selected. The numerical tests show that the trained CLSP-Net achieves an accuracy of 78.17 % for the test set of data collection J72T12G24.

The next best alternative forecasting method uses conditional probabilities and achieves an accuracy of 70.63 %. The remaining three forecasting methods achieve significantly worse results. While a rule-based approach still achieves an accuracy of up to 45.13 %, a random selection according to the probability distribution contained in the data generates an average accuracy of 33.13 % and a random selection according to a uniform distribution generates an average accuracy of only 25.05 %.

With the data collection J108T18G36, another data collection is created that contains problem instances 125 % larger than the data collection J72T12G24. It should be mentioned that all forecasting methods were not trained on the basis of such large problem instances. The numerical tests show that CLSP-Net achieves a comparable accuracy of 78.61 % without a new training phase. Moreover, the computation time for predicting a single problem instance for very large problem sizes is less than 2 seconds, so computation time can be saved compared to testing all heuristics regardless of the problem size. On the other hand, it can be seen that the generation of CLSP-Net (especially the generation of the training, validation and test data collection) takes a lot of time.

References

[1] L. Buschkühl, F. Sahling, S. Helber, and H. Tempelmeier, 'Dynamic capacitated lot-sizing problems: a classification and review of solution approaches', OR Spectrum, vol. 32, no. 2, pp. 231–261 (2010).

[2] X. Chu, F. Cai, C. Cui, M. Hu, L. Li, and Q. Qin, 'Adaptive recommendation model using meta-learning for population-based algorithms', Information Sciences, vol. 476, pp. 192–210 (2019).

[3] K. Copil, M. Wörbelauer, H. Meyr, and H. Tempelmeier, 'Simultaneous lotsizing and scheduling problems: a classification and review of models', OR Spectrum, vol. 39, no. 1, pp. 1–64 (2017).

[4] A. L. Dantas and A. T. R. Pozo, 'A Meta-Learning Algorithm Selection Approach for the Quadratic Assignment Problem', in 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2018).

[5] 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, vol. 2, no. 1, pp. 23–39 (1981).

[6] A. Dogramaci, J. C. Panayiotopoulos, and N. R. Adam, 'The Dynamic Lot-Sizing Problem for Multiple Items Under Limited Capacity', AIIE Transactions, vol. 13, no. 4, pp. 294–303, (1981).

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, pp. 801. MIT Press, Cambridge MA (2016).

[8] H.-O. Günther, 'Planning lot sizes and capacity requirements in a single stage production system', European Journal of Operational Research, vol. 31, no. 2, pp. 223–231 (1987).

[9] R. Jans and Z. Degraeve, 'Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches', European Journal of Operational Research, vol. 177, pp. 1855–1875 (2007).

[10] R. Jans and Z. Degraeve, 'Modeling industrial lot sizing problems: a review', International Journal of Production Research, vol. 46, no. 6, pp. 1619–1643 (2008).

[11] B. Karimi, S. M. T. Fatemi Ghomi, and J. M. Wilson, 'The capacitated lot sizing problem: a review of models and algorithms', Omega, vol. 31, no. 5, pp. 365–378 (2003).

[12] M. Karimi-Mamaghan, M. Mohammadi, P. Meyer, A. M. Karimi-Mamaghan, and E.-G. Talbi, 'Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art', European Journal of Operational Research, vol. 296, no. 2, pp. 393–422 (2022).

[13] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, 'Automated Algorithm Selection: Survey and Perspectives', Evolutionary Computation, vol. 27, no. 1, pp. 3–45 (2019).

[14] M. R. Lambrecht and H. Vanderveken, 'Heuristic Procedures for the Single Operation, Multi-Item Loading Problem', AIIE Transactions, vol. 11, no. 4, pp. 319–326 (1979).

[15] H. Meyr and M. Mann, 'A decomposition approach for the General Lotsizing and Scheduling Problem for Parallel production Lines', European Journal of Operational Research, vol. 229, no. 3, pp. 718–731 (2013).

[16] M. Wörbelauer, 'Simultaneous lotsizing and scheduling - extensions and solution approaches', Dissertation, University of Hohenheim (2018).

Mathematical Formulations and Reformulations

WGB 1.07 - Friday, 25/08 - 09:00-10:30

Formulations for the one-warehouse multi-retailer problem with production constraints

Agathe L'Hermite
École Polytechnique
agathe.lhermite@polytechnique.edu

Marilène Cherkesly, Matthieu Gruson
École des sciences de la gestion, Université du Québec à Montréal
{cherkesly.marilene, gruson.matthieu}@uqam.ca

Abstract

In this paper, we introduce, model and solve the one-warehouse multi-retailer problem with production constraints (OWMR-PC). In the OWMR-PC, we consider one warehouse that produces one type of item over a discrete and finite planning horizon. The items are transported to retailers which have to satisfy a known customer-demand. We explore different types of production constraints: 1) ordering is only permitted at a predefined set of periods, 2) limiting the number of production periods either through a maximum number of production periods or through a minimal or maximal number of periods between production. Those constraints mimic a situation where the length of time periods is different between the warehouse and the retailers. The objective consists of finding a solution which minimizes the operational costs, comprising a fixed production and order cost and an inventory holding cost, which respects the predefined set of constraints including the production constraints. We propose different ways to adapt the state-of-the-art formulations for the OWMR to the OWMR-PC. We conduct extensive computational experiments to show the limitations of each formulation and we derive appropriate managerial insights related to considering production constraints.

1 Introduction

In this work, we introduce, model and solve the one-warehouse multi-retailer with production constraints (OWMR-PC). The OWMR-PC is defined over a finite planning horizon of T periods, $1 \leq t \leq |T|$, and only one type of commodity (item) is considered. At each time period $t \in T$, a single warehouse can produce items incurring a setup cost (fixed production cost) denoted by f_t^0. A set of retailers C can then order items from the warehouse. Each retailer $c \in C$ must satisfy a known demand at each time period $t \in T$ denote by d_t^c. A setup cost (fixed ordering cost) denoted by f_t^c is incurred by retailer $c \in C$ when it orders from the warehouse at period $t \in T$. At the end of each period $t \in T$, items can be kept in inventory both at the warehouse and at the retailers incurring a holding cost of h_t^0 and $h_t^c, c \in C$, per item at the warehouse and at the retailers, respectively. Production constraints limit the periods when the warehouse can produce items. Two types of production constraints are considered: 1) producing only at a predefined set of periods, 2) limitations on the production periods which can be imposed through a maximum number of production periods, or through a minimal or a maximal number of periods between production. The problem consists of determining the number of items to produce at the warehouse and to order at the retailers at each time period in order to satisfy the demand at the retailers while respecting the production constraints and minimizing the total costs (i.e., setup and holding costs).

Production constraints are important from a practical stand-point. In fact, production requires additional workload for warehouses. Some days of the week or periods of the year may be impractical for production due to various events or restrictions. For example, it can be more complex to produce items over on Saturdays and Sundays, even though the retailers could pass orders on these days. Similarly, producing after some holidays (e.g., Chinese new year) might be more complex in terms of workload. Therefore, consider such constraints is important to understand the impacts both in terms of methodological developments as well as practical implications.

To the best of our knowledge, we are the first to study the OWMR-PC. This problem is closely related to the one-warehouse multi-retailer problem (OWMR) which has been studied in the literature and for which many mathematical models have been proposed [2, 3]. From a practical point of view, the multi-commodity formulation (MC) is the most efficient formulation found so far. Note that generally, authors consider that warehouses *order* at each time period rather than *produce*. To remove the confusion related to the *production constraints*, which could be seen as ordering constraints at the warehouse level, in the OWMR-PC, we say that the warehouse *produces* rather than *orders*. The OWMR-PC is also related to the joint replenishment problem (JRP) [1] which can been seen as a particular case of the OWMR problem where storage is not permitted at the warehouse.

2 Mathematical model

For conciseness reasons, we do not present a full mathematical model for the OWMR-PC. In the following, we explain how we adapt the multi-commodity formulation as introduced by [2] to consider the proposed production constraints. Let us recall that binary variables y_t^0 are equal to one if the warehouse produces item at period $t \in T$ and zero otherwise.

2.1 Fixed production periods

The first type of production constraints considers that production can only be done at predefined periods, i.e., production is forbidden at known time periods. We denote by $X \subseteq T$ the set of forbidden periods. We consider three methods to model fixed production periods. Method 1 consists of fixing the values of the y-variables by adding $y_t^0 = 0, \ \forall t \in X$. Method 2 consists of considering a high production setup cost at forbidden periods by setting $f_t^0 = M_t, \forall t \in X$, where M_t is a high value. We consider two ways to compute M_t: a naive value where $M_t = \infty, \forall t \in X$, and a value which relies on computing upper bounds on the costs, $\tilde{M}_t, \forall t \in X$. Method 3 consists of imposing additional constraints which model restrictions on the inventory between consecutive non-forbidden periods, through the echelon-stock concept.

2.2 Limitations on the production periods

The second type of production constraints considers that production can be done at any time period (no forbidden periods), but limits the production periods. Three possibilities are modeled. The first one consists of imposing a maximum number of production periods n which is imposed through

$$\sum_{t \in T} y_t^0 \leq n. \tag{1}$$

The second one consists of imposing a minimum gap of τ periods between production periods and can be imposed through

$$\sum_{k=t}^{t+\tau-1} y_k^0 \leq 1, \quad \forall\ 1 \leq t \leq |T| - \tau + 1. \tag{2}$$

The third one consists of imposing a maximum gap of τ periods between production periods and can be imposed through

$$\sum_{k=t}^{t+\tau-1} y_k^0 \geq 1, \quad \forall\ 1 \leq t \leq |T| - \tau + 1. \tag{3}$$

Note that we also study alternative ways to model Constraints (1), (2) and (3).

3 Computational results

All experiments were conducted on a Linux x86_64 machine equipped with an Intel Core i7-7700 3.60 GHz processor and 62Go of RAM. The code was implemented in Python 3.9.14 and CPLEX 22.12 was utilized. During all executions, a time

$\|X\|$	y_t^0	M_t		Add constraints
		∞	$\tilde{M_t}$	
16	2.27	4.46	3.97	4.77
18	1.97	4.88	3.88	4.66
19	1.71	5.31	3.43	4.40

Table 1: Solving time in seconds ($|T| = 25, |C| = 100$)

$\|X\|$	Setup costs (%)		Holding costs (%)		Total costs
	Retailers	Warehouse	Retailers	Warehouse	
16	30	17	39	14	161,773
18	25	11	39	25	183,693
19	18	7	40	35	213,822

Table 2: Repartition of the total costs ($|T| = 25, |C| = 100$)

limit of 7,200 seconds was enforced. Our preliminary results reaffirm that the multi-commodity formulation proposed by [2] is the most efficient. When imposing fixed production periods, fixing the y-variables proves to be the most efficient way, while considering a naive value of M_t is the least efficient. Table 1 presents the different methods' solving time for representative instances. An analysis on the cost repartition indicates that when we increase the number of forbidden production periods, the setup costs (and the number of setups) at the retailers decreases while the holding costs at the warehouse increase. Table 2 presents for a different number of forbidden production periods the setup and holding costs (in percentage) at the retailer and warehouse levels, as well as the total costs. Finally, we show that Constraint (1) performs generally well and gives similar results concerning the repartition of total costs.

References

[1] Arkin, E., Joneja, D., Roundy, R. Computational complexity of uncapacitated multi-echelon production planning problems. Operations Research Letters, 8, 61–66 (1989)

[2] Cunha, J. O., Melo, R. A. On reformulations for the one-warehouse multi-retailer problem. Annals of Operations Research, 238, 99–122, (2016)

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

Timed Route Approaches for Production Planning with Time Constraints

Benjamin Anthouard[1,2] Quentin Christ[2] Stéphane Dauzère-Pérès[1,3] Renaud Roussel[2]

[1]Mines Saint-Etienne, Univ Clermont Auvergne
CNRS, UMR 6158 LIMOS
F-13541 Gardanne, France
E-mail: b.anthouard@emse.fr, dauzere-peres@emse.fr

[2]STMicroelectronics Crolles
F-38926 Crolles, France
E-mail: quentin.christ@st.com@st.com, renaud.roussel@st.com

[3]Department of Accounting, Auditing and Business Analytics
BI Norwegian Business School
0484 Oslo, Norway

Abstract

Semiconductor manufacturing processes are probably the most complex manufacturing processes in the world. More recently, time constraints (TCs) have significantly increased the production management complexity [4]. Our problem is a multi-product, multi-step, multi-machine production planning problem with TCs, where the work-in-process, inventory and backlog costs are minimized. This work focuses on the integration of TCs and practical considerations in tactical production planning using the notion of Timed Routes (TR) and a column generation approach from the literature. In particular, the notion is extended to Machine Timed Routes (MTR). Three mathematical models are proposed with time constraints: Fixed lead time, TR and MTR. The models are compared using industrial instances and used to analyze the negative impacts of TCs on tactical plans.

1 Context, problem statement and motivations

Semiconductor manufacturing processes are probably the most complex manufacturing processes in the world. They are characterized by long cycle times, a very large number of operations (more than 1,000 for some product routes) that require hundreds of different machines to process a large volume of wafers constantly circulating in the manufacturing facility, called fabs. Time constraints (TCs), defined between two processing steps (consecutive or not), increase the complexity of managing semiconductor fabs. Time constraints might "cover" more than 20 operations and overlap or follow each other creating Time Constraint Tunnels (TCTs). In TCs and TCTs, lots must respect a maximum time between the two processing steps to ensure the quality of the wafers in the lots.

Time constraints tend to reduce the throughput of the fabs and need to be taken into account when determining tactical plans. Each product has a route, i.e. needs to perform a certain sequence of processing steps, and each step has a processing duration and can be performed by multiple machines. This problem is a multi-product, multi-step, multi-machine production planning problem. The goal is to minimize the work-in-process, inventory and backlog costs.

Mathematical programming methods can be used to solve production planning problems with limited capacity and fixed lead times [5]. As modeling with fixed lead times has limitations, in particular in correctly using production capacity, workload dependent lead times using clearing functions and simulations [1] have been proposed in the literature. Flexible lead times [3] have also been proposed.

This work focuses on the integration of time constraints and practical considerations in tactical production planning using the notion of timed routes and the column generation approach introduced in [2]. Moreover, industrial instances are used to validate the proposed approach. Timed routes are product routes with a time period assigned to each processing step. We extend the column generation approach of [2] to take industrial constraints, including time constraints, into account.

2 Solution approach

As the duration of critical time constraints is usually smaller than one day, the time period in the planning horizon considered should be at most one day. Due to the duration of some long processing steps, shorter periods can hardly be considered as they might make the modeling of machine capacity more difficult. Also, to better fit the industrial reality, when considering time constraints, detailed machines should be taken into account and not only workshops (set of machines) as in [2]. Indeed, as machines are different even in the same workshop (as machines can only perform a limited number of operations of products), they need to be considered independently. Industrially, some of the most complex TCTs to manage are TCTs in which some processing steps have only one eligible (or qualified) machine. In addition, as a wafer fab is never empty, the initial Work-In-Process (i.e. products already started and waiting in front of machines or being processed) should be considered. Three mathematical models are proposed.

First, we extended the fixed lead time model of [2] which ensures, by definition of the fixed lead times, that the time constraints are respected. However, the lack of flexibility of fixed lead times makes the model not relevant. Considering flexible lead times makes the model too large and thus too slow to solve with a standard solver.

To face this problem, [2] introduce the concept of timed routes. As already explained, a Timed Route (TR) is a route in which every processing step is assigned to a period. We extend this concept to Machine Timed Route (MTR), where the machine on which to process each step is also specified. Thus two different models are proposed: **(i)** A TR model where TR patterns are used to solve a linear programming model, and the quantity to process on each TR and the machines assigned to each step of the TR are optimized and **(ii)** A MTR model where MTR patterns are used to solve a linear programming model where "only" the quantity to be processed on each MTR is optimized. These models have been improved to also consider the initial inventories.

To solve these two models, the column generation approach of [2] is extended. An initial set of TR and MTR is generated using historical data, and new TR and MTR are generated by solving the pricing problem and using similar dominance properties as in [2]. In addition, time constraints are implemented as constraints to be respected when generating new timed routes and machine time routes. By extension, if a route respects all its TCs, then its TCTs are respected. However, this extension can be too hard for critical TCTs. For example, for a TCT that include TCs shorter than one period (i.e. with duration shorter than one period), the steps of each TC have to be performed in the same period. This means, by extension, that all the steps of the TCT have to be performed in the same period. This can be too constraining as a lot might actually enter a TC at the end of the tunnel only at the end of the period, or even in the next period. This is why a coefficient T^{relax} has been introduced to allow some TCs to be performed on multiple periods.

3 Numerical results

The three mathematical models have been compared using industrial instances. The numerical results obtained so far on three industrial instances will be presented in the workshop. They show that the MTR model outperforms the fixed lead time model and the TR model in terms of computational times, but also of solution quality (see Table 3). In terms of computational times, dual resolution seems to generally perform better than primal resolution. However, in the column generation approach, the primal resolution seems to outperform the dual resolution when the MTR model is used, as it manages to solve the problem at one iteration by starting from the solution obtained in the previous iteration and adding the new columns.

As expected, time constraints have a negative impact on the quality of the optimized production plans as they reduce the capacity to fulfill the demand. For some instances, large impacts could be observed on the satisfaction of the demand of some

	FLT Dual	TR Dual	MTR Primal
CPU time	18h	>48h	48h
Nb iterations	1	145	2 124
Gap to optimal MTR objective	868%	8%	0%
Not satisfied demand	6%	1.7%	1.7%
Nb created routes	-	4 351	39 106

Table 1: Comparison between solutions obtained with FLT model and column generation approached for TR and MTR models.

product with shorter (i.e. more constraining) TCs than the other products. On other instances, a limited impact on the demand is observed as the due dates seem to be already short. However, different production plans are proposed that satisfy the TCs when solving the TR and MTR models.

4 Conclusions and perspectives

The timed route model and the column generation approach proposed in [2] have been extended at the machine level considering periods of one day and to consider additional industrial constraints, including time constraints. Computational experiments on industrial instances show that the MTR model performs very well.

This work has multiple perspectives. The initialization of the TR model and the MTR model could be improved by proposing a better set of initial time routes. In addition, considering periods of different lengths should be relevant to optimize production plans on longer horizons. This is particularly relevant in semiconductor manufacturing where products have very long cycle times.

References

[1] Dieter Armbruster and Reha Uzsoy. Continuous dynamic models, clearing functions, and discrete-event simulation in aggregate production planning. In *New Directions in Informatics, Optimization, Logistics, and Production*, pages 103–126. INFORMS, 2012.

[2] Sébastien Beraudy, Nabil Absi, and Stéphane Dauzère-Pérès. Timed route approaches for large multi-product multi-step capacitated production planning problems. *European Journal of Operational Research*, 300(2):602–614, 2022.

[3] M Chen, SC Sarin, and Andy Peake. Integrated lot sizing and dispatching in wafer fabrication. *Production Planning and Control*, 21(5):485–495, 2010.

[4] Alexandre Lima, Valeria Borodin, Stéphane Dauzère-Pérès, and Philippe Vialletelle. A sampling-based approach for managing lot release in time constraint tunnels in semiconductor manufacturing. *International Journal of Production Research*, 59(3):860–884, 2021.

[5] JM Spitter, Cor AJ Hurkens, AG De Kok, Jan Karel Lenstra, and Ebbe G Negenman. Linear programming models with planned lead times for supply chain operations planning. *European Journal of operational research*, 163(3):706–720, 2005.

The integrated lot-sizing and storage assignment problem

Gislaine Mara Melega
HEC Montréal and CIRRELT, Canada H3T 2A7 QC, Canada
gislaine-mara.melega@hec.ca

Chi Xu
HEC Montréal and CIRRELT, Canada H3T 2A7 QC, Canada
chi.xu@hec.ca

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

Julie Paquette
HEC Montréal and CIRRELT, Canada H3T 2A7 QC, Canada
julie.paquette@hec.ca

Abstract

This study addresses the integration of the lot-sizing and storage assignment problems. Traditional lot-sizing problems have been extensively studied, but research has only recently paid attention to the assignment of items to inventory locations. In our problem, the storage space is divided into separate locations, and inventory is assigned based on specific conditions. Relocation of inventory is possible if needed. Apart from traditional cost elements, we consider additional inventory-related costs like fixed storage cost, handling cost, and relocation cost. We propose a general mathematical model and a transportation reformulation. To solve the integrated problem, we design heuristics that split it into smaller subproblems and solve them sequentially. Computational experiments evaluate the behavior of the integration and different solution approaches. We also analyze the impact of key input parameters on the solution. By studying the interaction between lot-sizing and storage assignment, we aim to better approach the complexities of real-world scenarios.

1 Introduction

Considering the dynamic lot-sizing problem as one of the most studied production planning problems, we study the complex case that arises when multiple capacitated storage locations are available. In lot-sizing literature, limitations arise with respect to the production capacity, as well as the storage capacity, since inventory cannot accumulate infinitely. One of the most obvious limitations is the space used to store the inventory. Some of the studies on lot-sizing problems have addressed storage capacity, although still allowing an unlimited production capacity ([7, 2, 6]), while others consider both production and storage capacities ([1, 4, 9]).

An example of the multiple capacitated storage locations is in the storage of liquid products into tanks, where the typical approach of considering a single global shared storage capacity is no longer valid since the empty space in a tank cannot be used to contain a second type of product because different liquids cannot be mixed. When different types of products need to be stored in multiple locations, we also take into account whether products can be stored together or not. In some cases certain products cannot be stored together due to incompatibilities. Examples of these types of restrictions are commonly seen in the chemical industry ([4]), where special incompatibility restrictions have to be strictly obeyed. Similarly, we have to consider the compatibility between a specific product and a location. Some items may be stored in multiple locations, whereas others require a specific location.

The presence of multiple locations leads to the consideration of different costs, in addition to the traditional holding cost. We separately incorporate the cost of the activities related to the assignment of items to specific capacitated storage locations: the fixed storage location costs ([8, 3, 9]), which is related to the use of a specific location to hold inventory in a specific period; the variable handling costs that specifically reflects the effort to move a unit from the production line to the assigned storage location; and the relocation costs that is the manpower and energy required by the movements needed to relocate the items from one storage location to another.

2 Mathematical Model and Solution Approaches

In this study, we propose an integrated capacitated multi-item lot-sizing and storage assignment problem with multiple capacitated storage locations. We consider a cost structure that incorporates the new cost elements (fixed storage costs, handling costs and relocation costs), along with the traditional costs from the lot-sizing problems (setup costs, production costs and inventory costs), aiming to balance the multiple trade-offs among the different costs. The problem takes into account a finite planning horizon with dynamic deterministic demand. In the production environment, in a given period, multiple items can be produced using the same resource. Furthermore, multiple capacitated storage locations are available to store items, and the storage capacity, variable and fixed storage costs are location-dependent. In this way, the produced items can be either directly shipped to the clients to satisfy demand or stored in the storage locations. On the other hand, the demands can be fulfilled with items coming either directly from the production line in the same period or from the inventory accumulated over previous periods. From period to period, the inventory of items can be moved from one location to another in order to maintain feasibility by respecting compatibility issues or with the purpose of economizing space and avoiding unnecessary costs.

As a result of the different characteristics of items and storage locations, it is necessary to track the specific assignment of the inventory of each item in each storage location. For this, we propose a mathematical formulation for the problem and we reformulate it as a transportation problem, in order to obtain better lower bounds ([5]). Therefore, an economic production and storage plan should be designed taking into account the production and holding costs, the newly introduced fixed and variable cost, compatibility requirements of the multiple storage locations, and production and storage restrictions, while fulfilling the demand of each item over a series of discrete time periods.

To solve the integrated lot-sizing and storage assignment problem, we design several heuristics that are based on the idea of splitting the integrated problem into smaller subproblems, which are then solved sequentially. Several decompositions are proposed. A lower bound is also obtained from one of the heuristics. We perform an extensive computational study to assess the impact of the integration between the decisions, as well as the behavior of the different solution approaches. We also perform a sensitivity analysis in order to better understand the trade-offs of some key input parameters on the solution and provide meaningful managerial insights to obtain a more efficient production planning and assignment of items into storage locations.

The results demonstrate that the proposed heuristics are highly effective in finding feasible solutions that are very close to the best solutions, with an average gap of 0.81%. On average, they achieve this with 97% less computational time compared to solving the full mathematical model, even for large instances of the problem. In comparison with benchmark heuristics, some versions of the heuristics are able to find better solutions in a considerable shorter computational time, highlighting the benefit of using more specific heuristics. The results from the sensitivity analysis highlights the influence of compatibility levels on the problem complexity. Limited item-item compatibility adds considerably to the complexity, whereas limited item-location compatibility tights the constraints reducing computational time.

Acknowledgement(s)

The authors gratefully acknowledge the support of the Canadian Social Sciences and Humanities Research Council (SSHRC 430-2017-00894) and the HEC Montréal Research Chair in Supply Chain Operations Planning.

References

[1] A. Akbalik, B. Penz, and C. Rapine. Capacitated lot sizing problems with inventory bounds. *Annals of Operations Research*, 229(1):1 – 18, 2015.

[2] A. Akbalik, B. Penz, and C. Rapine. Multi-item uncapacitated lot sizing problem with inventory bounds. *Optimization Letters*, 9(1):143 – 154, 2015.

[3] A. Atamtürk and S. Küçükyavuz. An o(n2) algorithm for lot sizing with inventory bounds and fixed costs. *Operations Research Letters*, 36(3):297 – 299, 2008.

[4] A. L. Cunha, M. O. Santos, R. Morabito, and A. B. Póvoa. An integrated approach for production lot sizing and raw material purchasing. *European Journal of Operational Research*, 269(3):923 – 938, 2018.

[5] J. Krarup and O. Bilde. Plant location, set covering and economic lot size: An 0 (mn) - algorithm for structured problems. In L. Collatz, G. Meinardus, and W. Wetterling, editors, *Numerische Methoden bei Optimierungsaufgaben Band 3*, volume 36 of *International Series of Numerical Mathematics*, pages 155 – 180. Birkhäuser Basel, 1977.

[6] R. A. Melo and C. C. Ribeiro. Formulations and heuristics for the multi-item uncapacitated lot-sizing problem with inventory bounds. *International Journal of Production Research*, 55(2):576 – 592, 2017.

[7] S. Minner. A comparison of simple heuristics for multi-product dynamic demand lot-sizing with limited warehouse capacity. *International Journal of Production Economics*, 118(1):305 – 310.

[8] M. Van Vyve and F. Ortega. Lot-sizing with fixed charges on stocks: the convex hull. *Discrete Optimization*, 1(2):189 – 203, 2004.

[9] G. Zhang, X. Shang, F. Alawneh, Y. Yang, and T. Nishi. Integrated production planning and warehouse storage assignment problem: An iot assisted case. *International Journal of Production Economics*, 234:108058, 2021.

Stochastic Inventory Management

Aula Maxima - Friday, 25/08 - 11:00-12:30

Robust Models for Remanufacturing with Multiple Quality Classes

Farzad Shams
Dept. of Management Science, University of Strathclyde, Glasgow, UK
farzad.shams@strath.ac.uk

Agostinho Agra
Department of Mathematics and CIDMA, University of Aveiro, Aveiro, Portugal
aagra@ua.pt

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

Abstract

In this talk, we discuss the robust multi-stage lot-sizing problem, when the returned or collected products (cores) are classified into multiple quality classes (RML-MQ). For example, end-of-lease or returned laptops from different sources and channels could be remanufactured to a certain acceptable quality level and configuration for sale, although the amount of effort for bringing any two random laptops to a like-new state could be quite different. In our problem, we consider N time periods, a single type of product, and random demand, and backlogging is also allowed. We discuss various robust optimization models and approaches for the problem, including decomposition and dualization, as well as various challenges in handling adjustable variables. We conclude with some preliminary results and future perspectives.

1 Introduction

In this paper, we study the robust multi-stage lot-sizing problem, when the returned or collected products (cores) are classified into multiple quality classes (RML-MQ). A typical example with such setting would be the IBM's remanufacturing facility in Raleigh N.C., where the firm receives end-of-lease or returned laptops from different sources and channels ([3]). Each returned laptop could be remanufactured to a certain acceptable quality level and configuration before being put in the market for sale again. In a given inventory of returned laptops, the amount of effort required for bringing any two laptops to a like-new state could be quite different. While one laptop may just require a thorough cleaning and formatting the hard drive, another laptop may require a few new parts, e.g., a worn out screen panel to be replaced with a new one, or a new memory card to be installed to replace a faulty one. This directly effects the time and cost of remanufacturing, creating an important issue that has to be considered in the production planning.

We consider a finite horizon of N time periods with a single type of product. In each period, the manufacturer receives a random amount of returned products, which then are graded and grouped into Q different quality classes. We assume that the returned products are sorted before any decision is made, and that the cost of sorting activities to be a sunk cost, as they do not directly impact the problem structure. The manufacturer has the option to remanufacture the available cores to fulfill the demand or keep them in stock to be remanufactured in a future period, and backlogging of demand is also allowed, as commonly observed in the literature (e.g., [1, 4, 5]). The unit manufacturing cost is higher than the remanufacturing cost of any type of core (not only for remanufacturing to be a reasonable option but also in line with practical examples), and graded cores can be disposed at a certain cost.

The problem is to find the optimal values for production of new products, and the amounts of different quality cores to remanufacture or salvage in each time period, to minimize the total costs of production, inventory/backlogging, and disposal. We also assume that the customers are indifferent between manufactured and remanufactured products.

2 Deterministic Model

Let x_t^m be the amount of items manufactured, x_t^q be the amount of items of quality class q remanufactured, and s_t^q be the amount of items of quality class q salvaged, in period t. Also let y_t be binary variables to indicate whether a joint setup has taken place or not. Then, a deterministic MIP model can be presented, in a similar fashion to [2], as follows.

$$\min \sum_{t=1}^{N} (f_t y_t + c_0 x_t^m + \sum_{q=1}^{Q} (c_q x_t^q + c_{Q+1} s_t^q) + I_t^s + \sum_{q=1}^{Q} I_t^q) \tag{1}$$

$$\text{s.t. } I_t^s \geq h^s \sum_{i=1}^{t} (x_i^m + \sum_{q=1}^{Q} x_i^q - d_i), \qquad t \in N \tag{2}$$

$$I_t^s \geq -b \sum_{i=1}^{t} (x_i^m + \sum_{q=1}^{Q} x_i^q - d_i), \qquad t \in N \tag{3}$$

$$I_t^q \geq h^q \sum_{i=1}^{t} (r_i^q - x_i^q - s_i^q), \qquad q \in Q, t \in N \tag{4}$$

$$\sum_{i=1}^{t} (r_i^q - s_i^q - x_i^q) \geq 0 \qquad q \in Q, t \in N \tag{5}$$

$$x_t^m + \sum_{q=1}^{Q} x_t^q \leq M_t y_t, \qquad t \in N \tag{6}$$

$$y_t \in \{0, 1\}, \qquad t \in N \tag{7}$$

$$x_t^m, x_t^q, s_t^q \geq 0, \qquad t \in N. \tag{8}$$

The objective is to minimize the total cost incurred due to setup, manufacturing, remanufacturing, disposal, inventory, and shortage costs.

3 Uncertainty

Firstly, we introduce the following notation:

$$d^t := (d_1, ..., d_t) \quad \text{and} \quad r^{q,t} := (r_1^q, ..., r_t^q)$$

This notation allows us to define variables $x_t^m(d^{t-1}), x_t^q(r^{q,t-1})$, and $s_t^q(r^{q,t-1})$ as functions of the past data d^t and $r^{q,t}$, and hence, an adjustable multi-stage robust model. In the talk, we will discuss several affine decision rules, and some preliminary computational results.

Acknowledgments

The work of the third author is supported by the Air Force Office of Scientific Research under award number FA9550-18-1-7003.

References

[1] Ö.N. Attila, A. Agra, K. Akartunalı, A. Arulselvan, Robust Formulations for Economic Lot-Sizing Problem with Remanufacturing, European Journal of Operational Research, 288(2):496-510 (2021).

[2] Ö.N. Attila, A. Agra, K. Akartunalı, A. Arulselvan, Robust Two-level Multi-component Lot-sizing Problem with Remanufacturing, Working paper (2023).

[3] M. Denizel, M. Ferguson, G.C. Souza, Multiperiod Remanufacturing Planning with Uncertain Quality of Inputs, IEEE Transactions on Engineering Management, 57(3):394-404 (2009).

[4] C-T. See, M. Sim, Robust Approximation to Multiperiod Inventory Management, Operations Research, 58(3):583-594 (2010).

[5] Z. Tao, S.X. Zhou, Approximation Balancing Policies for Inventory Systems with Remanufacturing, Mathematics of Operations Research, 39(4):1179-1197 (2014).

Uncapacitated lot-sizing problems with remanufacturing and two sale markets

Siao-Leu Phouratsamay
Univ. Grenoble Alpes, CNRS, Grenoble INP G-SCOP, 38000 Grenoble, France
siao.phouratsamay@grenoble-inp.fr

Bernard Penz
Univ. Grenoble Alpes, CNRS, Grenoble INP G-SCOP, 38000 Grenoble, France
barnard.penz@grenoble-inp.fr

Abstract

The remanufacturing process consists in recovering returned products by replacing or repairing components. In this work, we consider an uncapacitated lot-sizing problem with remanufacturing and two types of markets. The demand of new products is only satisfied by manufactured products, whereas the demand of second-hand products is only satisfied by the remanufacturing of returned products. We develop polynomial dynamic programming algorithms for two cases, where the returned products can be disposed or not.

1 Introduction

The remanufacturing is the process of recovering returned products (returns) by replacing or repairing components with a bad quality. The research on production with remanufacturing has been growing over the decades, especially on lot-sizing problems [9].

The lot-sizing problem with remanufacturing was first studied in [5]. The authors propose a polynomial algorithm when the costs are linear and prove that the problem is NP-complete for general concave costs. Polynomial time dynamic programming algorithms has been developed to solve special cases of the problem [1, 6, 10], where other papers focused on formulations of the problem [3, 8]. Recently, Piñeyro and Veira [7] introduce heterogeneous quality for the returns. They provide complexity results, algorithms to solve different cases of the problem and an experimental analysis to evaluate the relevance of using the returns. No paper in the literature distinguishes the demand of new and second-hand items in the lot-sizing problem with remanufacturing.

In the lot-sizing litterature, remanufactured products are considered as equivalent to new products, and can satisfy a part of the demand. However, it is not practical to consider them as new products since the market price, the clients and the marketing strategy are different [2, 4]. This motivates us to study the lot-sizing problem with remanufacturing and two sale markets, one for new products and another one for remanufactured products.

2 Problem definition

The planning horizon includes T discrete periods. The demand of new and second-hand products are denoted by D_t^n and D_t^s respectively. At the beginning of period t, R_t returns arrive and are available for remanufacturing.

Producing a new product (resp. a second-hand product) induces a cost of p_t^n (resp. p_t^s) in period t. Carrying a new product from period t to $t+1$ induces a unit holding cost of h_t^n (h_t^s for second-hand products and h_t^r for returns). At the end of a period t, returns which are not used to produce second-hand products can either be disposed or be placed in the inventory until the next period. The unit cost to dispose the returns is denoted by p_t^r.

We assume that the fixed setup cost f_t in period t is joint by the new and the second-hand products. Note that if the setup costs are not joint, the problem will consist in solving two independent uncapacitated lot-sizing problems.

Moreover, we assume that $D_t^n > 0$, $D_t^s > 0$, $h_t^r < h_t^s < h_t^n$.

The variables x_t^n and x_t^s denote the quantity of new and second-hand items produced in period t. The disposal quantity in period t is given by x_t^r. The inventory level of new products, second hand products and returns at the end of period t are noted s_t^n, s_t^s, s_t^r. The binary variable y_t indicates if a production of new or second-hand products occurs in period t.

The formulation of the uncapacitated lot-sizing problem with remanufacturing and two sale markets is as follows:

$$
\begin{array}{rcll}
\text{Minimize } z & = & \sum_{t=1}^{T}(f_t y_t + p_t^s x_t^s + p_t^n x_t^n + p_t^r x_t^r + h_t^s s_t^s + h_t^n s_t^n + h_t^r s_t^r) & \\
\text{s.t.} & & & \\
s_t^r & = & s_{t-1}^r + R_t - x_t^r - x_t^s & \forall t \in \{1, \cdots, T\} \\
s_t^s & = & s_{t-1}^s + x_t^s - D_t^s & \forall t \in \{1, \cdots, T\} \\
s_t^n & = & s_{t-1}^n + x_t^n - D_t^n & \forall t \in \{1, \cdots, T\} \\
x_t^s + x_t^n & \leq & M y_t & \forall t \in \{1, \cdots, T\} \\
x_t^r, x_t^s, x_t^n, & \in & \mathbb{R}^+ & \forall j \in \{1, \cdots, T\} \\
s_t^r, s_t^s, s_t^n, & \in & \mathbb{R}^+ & \forall j \in \{1, \cdots, T\} \\
y_t & \in & \{0, 1\} & \forall t \in \{1, \cdots, T\}
\end{array}
$$

3 Remanufacturing with joint setups and no disposal (ULSR-2m-nd)

In this section, we assume that the returns cannot be disposed. We propose an $\mathcal{O}(T^2)$ dynamic programming algorithm to solve this problem.

The ZIO property can be generalized to our problem as follows.

Proposition 1 *At each period t, $(x_t^s + x_t^n)(s_{t-1}^s + s_{t-1}^n) = 0$.*

A subplan $S_{u,v}$ is such that the stock is null only in periods $u-1$ and $v-1$ and for each period t such that $u \leq t \leq v-2$, we must have $s_t^s + s_t^n > 0$. Using the definition of a subplan and Proposition 1, the following proposition holds.

Proposition 2 *In a subplan $S_{u,v}$, the period u is the unique production period for both new and second-hand products, i.e. $x_u^s + x_u^n > 0$.*

From Propositions 1 and 2, we develop a polynomial time dynamic algorithm by showing that an optimal solution can be decomposed into a succession of independent subplans.

4 Remanufacturing with joint setups and disposal (ULSR-2m-d)

In this section, the returns can be disposed if they are not used to satisfy the second-hand demand. An $\mathcal{O}(T^6)$ dynamic programming algorithm is developed to solve this problem.

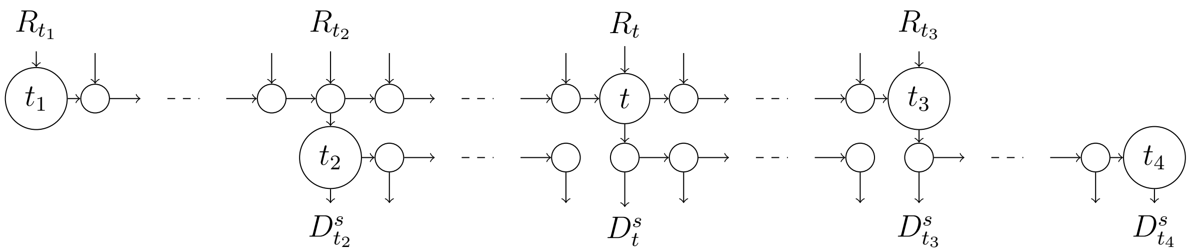


Figure 1: Structure of a separation-block

The idea of the algorithm is to decompose an optimal solution into independent block $\mathcal{B}(t_1, t_2, t_3, t_4)$ (see Figure 1) where t_1 is the unique period before t_3 with a positive disposal quantity, $t_2 \geq t_1$ is the first period with a positive production $(x_{t_2}^s +$

$x_{t_2}^n > 0$), and $t_3 \geq t_2$ is the last period of the block with a positive production quantity. We proved that the cost of an optimal solution for the block $\mathcal{B}(t_1, t_2, t_3, t_4)$ can be calculated in $\mathcal{O}(T^2)$. As the number of blocks are in $\mathcal{O}(T^4)$, the resulting dynamic program is in $\mathcal{O}(T^6)$.

References

[1] A. Arulselvan, K. Akartunalı, and W. van den Heuvel. Economic lot-sizing problem with remanufacturing option: complexity and algorithms. *Optimization Letters*, 16(2):421–432, 2022.

[2] A. Atasu, M. Sarvary, and L. N. Van Wassenhove. Remanufacturing as a marketing strategy. *Management Science*, 54(10):1731–1746, 2008.

[3] Ö. N. Attila, A. Agra, K. Akartunalı, and A. Arulselvan. Robust formulations for economic lot-sizing problem with remanufacturing. *European Journal of Operational Research*, 288(2):496–510, 2021.

[4] P. Couzon, Y. Ouazene, and F. Yalaoui. Joint optimization of dynamic pricing and lot-sizing decisions with nonlinear demands: Theoretical and computational analysis. *Computers & Operations Research*, 115:104862, 2020.

[5] B. Golany, J. Yang, and G. Yu. Economic lot-sizing with remanufacturing options. *IIE Transactions*, 33(11):995–1004, 2001.

[6] Z. Pan, J. Tang, and O. Liu. Capacitated dynamic lot sizing problems in closed-loop supply chain. *European Journal of Operational Research*, 198(3):810–821, 2009.

[7] P. Piñeyro and O. Viera. The economic lot-sizing problem with remanufacturing and heterogeneous returns: formulations, analysis and algorithms. *International Journal of Production Research*, 60(11):3521–3533, 2022.

[8] M. J. Retel Helmrich, R. Jans, W. van den Heuvel, and A. P. Wagelmans. Economic lot-sizing with remanufacturing: complexity and efficient formulations. *IIE Transactions*, 46(1):67–86, 2014.

[9] E. Suzanne, N. Absi, and V. Borodin. Towards circular economy in production planning: Challenges and opportunities. *European Journal of Operational Research*, 287(1):168–190, 2020.

[10] R. H. Teunter, Z. P. Bayindir, and W. van den Heuvel. Dynamic lot sizing with product returns and remanufacturing. *International Journal of Production Research*, 44(20):4377–4400, 2006.

First results on integrated stochastic lot sizing and rework planning

Pierre Kohlmann
RPTU Kaiserslautern-Landau, RPTU School of Business and Economics, Chair of Production Management, Germany
pierre.kohlmann@rptu.de

Florian Sahling
RPTU Kaiserslautern-Landau, RPTU School of Business and Economics, Chair of Production Management, Germany
florian.sahling@rptu.de

Abstract

In practice, an imperfect production process may result in a proportion of defective products that cannot be used to meet demand. The possibility of reworking defective products seems economically and environmentally reasonable. Typically, there is no ex-ante knowledge of the exact proportion of defective products. Therefore, we propose a nonlinear model formulation for integrated stochastic lot sizing and rework planning with random proportion of defective products. A sample average approach is used to approximate the nonlinear model. We apply a multistage stochastic programming approach that allows adjustments of future production and rework quantities to respond to specific realizations of the random proportion of defective products. First numerical results are presented that show the performance of the proposed approach.

1 Introduction

Production processes are often imperfect, and the result contains both defect-free and defective products. The latter cannot be used to satisfy demand; see [3]. Neglecting this aspect can lead to unsatisfied demand for defect-free products. Therefore, it is necessary to consider an imperfect production process in production planning. It is often possible to rework defective products. These products can then be used to meet demand. Typically, reworking defective products consumes less capacity and costs less than producing new products. In addition, waste is reduced; see [2]. Thus, reworking defective products seems economically and environmentally reasonable.

There is often no ex ante knowledge of the exact proportion of defective items within a production lot. An underestimation or overestimation has a direct impact on the planned lot sizes. An underestimation, for example, may lead to backlogs. Therefore, the randomness of the proportion of defective products must be explicitly considered.

2 Problem description

Figure 1 illustrates the considered planning situation.

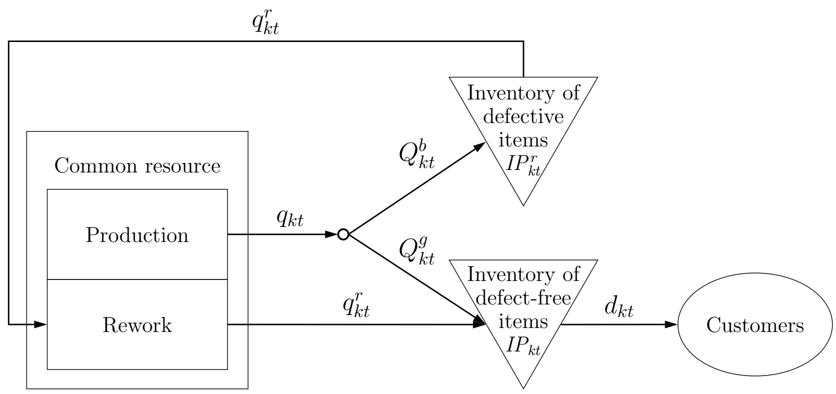


Figure 1: Integrated production and rework planning (cf. [4] and [2])

An imperfect production process produces a random proportion of defective products. Thus, the production quantity q_{kt} must be divided into defect-free Q^g_{kt} and defective products Q^b_{kt}. Production and rework are carried out by the same resource. The capacity of the resource can be increased by using overtime. However, the use of overtime is limited. Production and rework require separate setup operations that cause specific setup costs and times. In addition, production and rework of one unit entails variable costs and times. Both defect-free and defective products can be held in stock IP_{kt} and IP^r_{kt}. All defective products can be transformed into defect-free products, i.e., the rework process is perfect and the rework quantity q^r_{kt} can be used to satisfy the demand d_{kt}. The demand must be satisfied according to a δ-service level. The resulting nonlinear generic model is approximated using a sample average approach.

3 Flexible planning approach

To deal with the randomness of the proportion of defective products, we propose a multistage stochastic programming approach following the static-dynamic uncertainty strategy of [1]. In the first stage, setup decisions are determined. The first stage decisions are fixed for the entire planning horizon. In the following stages, period-specific adjustments of production and rework quantities are allowed. Each of the following stages corresponds to a single period t. In each period, production takes place first, followed by a rework process. If the production quantities are realized, the proportion of defective products becomes known. Based on this new information, the quantities of the following rework process can be adjusted. Simultaneously, production and rework quantities can be adjusted for all future periods.

4 First numerical results

We analyzed the performance of the proposed approach in a simulation-based analysis. We generated different realizations of the random proportion of defective products. For each realization, we applied the flexible planning approach. Based on fixed setup decisions, production and rework quantities of future periods can be adjusted after the proportion of defective items is known. We compared the results of the flexible approach with the results of a robust planning approach. In the robust planning approach, no adjustments to production and rework quantities are allowed. The simulation-based analysis shows that the flexible planning approach outperforms the robust planning approach.

References

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

[2] Goerler, A., and Voß, S., Dynamic lot-sizing with rework of defective items and minimum lot-size constraints, International Journal of Production Research, 54(8): 2284-2297 (2016).

[3] Goerler, A., Lalla-Ruiz, E., and Voß, S., Late acceptance hill-climbing matheuristic for the general lot sizing and scheduling problem with rich constraints, Algorithms, 13(6): 138 (2020).

[4] Inderfurth, K., Lindner, G., and Rachaniotis, N. P., Lot sizing in a production system with rework and product deterioration, International Journal of Production Research, 43(7): 1355-1374 (2005).