Lot sizing with setup carryover and crossover

Márcio Antônio Ferreira Belo Filho

Dimensionamento de lotes com preservação da
preparação total e parcial

Márcio Antônio Ferreira Belo Filho

Data de Depósito:

Assinatura: _____

Lot sizing with setup carryover and crossover¹

Márcio Antônio Ferreira Belo Filho

***Advisor:* Profa. Dra. Franklina Maria Bragion de Toledo**

***Co-Advisor:* Prof. Dr. Bernardo Sobrinho Simões Almada Lobo**

Doctoral dissertation submitted to the *Instituto de Ciências Matemáticas e de Computação* - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *EXAMINATION BOARD PRESENTATION COPY.*

USP – São Carlos
November 2014

¹ This work was financially supported by FAPESP (grant 2010/06901-1).

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

B4521 Belo Filho, Márcio Antônio Ferreira
 Lot sizing with setup carryover and crossover /
 Márcio Antônio Ferreira Belo Filho; orientadora
 Franklina Maria Bragion Toledo; co-orientador
 Bernardo Sobrinho Simões Almada-Lobo. -- São
 Carlos, 2014.
 132 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2014.

 1. Pesquisa Operacional. 2. Otimização
 Combinatória. 3. Planejamento da Produção. I. Toledo,
 Franklina Maria Bragion, orient. II. Almada-Lobo,
 Bernardo Sobrinho Simões, co-orient. III. Título.

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Dimensionamento de lotes com preservação da preparação total e parcial¹

Márcio Antônio Ferreira Belo Filho

***Orientadora:* Profa. Dra. Franklina Maria Bragion de Toledo**
***Co-Orientador:* Prof. Dr. Bernardo Sobrinho Simões de Almada Lobo**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA*.

USP – São Carlos
Novembro de 2014

¹ Este trabalho foi financiado pela FAPESP (processo 2010/06901-1).

Abstract

Production planning problems are of paramount importance within supply chain planning, supporting decisions on the transformation of raw materials into finished products. Lot sizing in production planning refers to the tactical/operational decisions related to the size and timing of production orders to satisfy a demand. The objectives of lot-sizing problems are generally economical-related, such as saving costs or increasing profits, though other aspects may be taken into account such as quality of the customer service and reduction of inventory levels. Lot-sizing problems are very common in production activities and an efficient planning of such activities gives the company a clear advantage over concurrent organizations. To that end it is required the consideration of realistic features of the industrial environment and product characteristics. By means of mathematical modelling, such considerations are crucial, though their inclusion results in more complex formulations. Although lot-sizing problems are well-known and largely studied, there is a lack of research in some real-world aspects.

This thesis addresses two main characteristics at the lot-sizing context: (a) setup crossover; and (b) perishable products. The former allows the setup state of production line to be carried over between consecutive periods, even if the line is not yet ready for processing production orders. The latter characteristic considers that some products have fixed shelf-life and may spoil within the planning horizon, which clearly affects the production planning. Furthermore, two types of perishable products are considered, according to the duration of their lifetime: medium-term and short-term shelf-lives. The latter case is tighter than the former, implying more constrained production plans, even requiring an integration with other supply chain processes such as distribution planning.

Research on stronger mathematical formulations and solution approaches for lot-sizing problems provides valuable tools for production planners. This thesis focuses on the development of mixed-integer linear programming (*MILP*) formulations for the lot-sizing problems considering the aforementioned features. Novel modelling techniques are introduced, such as the proposal of a disaggregated setup variable and the consideration of lot-sizing instead of batching decisions in the joint production and distribution planning problem. These formulations are subjected to computational experiments in state-of-the-art *MILP*-solvers. However, the inherent complexity of these problems may require problem-driven solution approaches. In this thesis, heuristic, metaheuristic and matheuristic (hybrid exact and heuristic) procedures are proposed. A lagrangean heuristic addresses the capacitated lot-sizing problem with setup carryover and perishable products. A novel dynamic programming procedure is used to achieve the optimal solution of the uncapacitated single-item lot-sizing problem with setup carryover and perishable item. A heuristic, a fix-and-optimize procedure and an adaptive large neighbourhood search approach are

proposed for the operational integrated production and distribution planning. Computational results on generated set of instances based on the literature show that the proposed methods yields competitive performances against other literature approaches.

Resumo

Problemas de planejamento da produção são de suma importância no planejamento da cadeia de suprimentos, dando suporte às decisões da transformação de matérias-primas em produtos acabados. O dimensionamento de lotes em planejamento de produção é definido pelas decisões tático-operacionais relacionadas com o tamanho das ordens de produção e quando fabricá-las para satisfazer a demanda. Os objetivos destes problemas são geralmente de cunho econômico, tais como a redução de custos ou o aumento de lucros, embora outros aspectos possam ser considerados, tais como a qualidade do serviço ao cliente e a redução dos níveis de estoque. Problemas de dimensionamento de lotes são muito comuns em atividades de produção e um planejamento eficaz de tais atividades, estabelece uma clara vantagem à empresa em relação à concorrência. Para este objetivo, é necessária a consideração de características realistas do ambiente industrial e do produto. Para a modelagem matemática do problema, estas considerações são cruciais, embora sua inclusão resulte em formulações mais complexas. Embora os problemas de dimensionamento de lotes sejam bem conhecidos e amplamente estudados, várias características reais importantes não foram estudadas.

Esta tese aborda, no contexto de dimensionamento de lotes, duas características muito relevantes: (a) preservação da preparação total e parcial; e (b) produtos perecíveis. A primeira permite que o estado de preparação de uma linha de produção seja mantido entre dois períodos consecutivos, mesmo que a linha de produção ainda não esteja totalmente pronta para o processamento de ordens de produção. A última característica determina que alguns produtos tem prazo de validade fixo, menor ou igual do que o horizonte de planejamento, o que afeta o planejamento da produção. Além disso, de acordo com a duração de sua vida útil, foram considerados dois tipos de produtos perecíveis: produtos com tempo de vida de médio e curto prazo. O último caso resulta em um problema mais apertado do que o anterior, o que implica em planos de produção mais restritos. Isto pode exigir uma integração com outros processos da cadeia de suprimentos, tais como o planejamento de distribuição dos produtos acabados.

Pesquisas sobre formulações matemáticas mais fortes e abordagens de solução para problemas de dimensionamento de lotes fornecem ferramentas valiosas para os planejadores de produção. O foco da tese reside no desenvolvimento de formulações de programação linear inteiro-mistas (*MILP*) para os problemas de dimensionamento de lotes, considerando as características mencionadas anteriormente. Novas técnicas de modelagem foram introduzidas, como a proposta de variáveis de preparação desagregadas e a consideração de decisões de dimensionamento de lotes ao invés de decisões de agrupamento de ordens de produção no problema integrado de planejamento de produção e distribuição. Estas formulações foram submetidas a experimentos computacionais em *MILP*-solvers de

ponta. No entanto, a complexidade inerente destes problemas pode exigir abordagens de solução orientadas ao problema. Nesta tese, abordagens heurísticas, metaheurísticas e matheurísticas (híbrido de métodos exatos e heurísticos) foram propostas para os problemas discutidos. Uma heurística lagrangeana aborda o problema de dimensionamento de lotes com restrições de capacidade, preservação da preparação total e produtos perecíveis. Um novo procedimento de programação dinâmica é utilizado para encontrar a solução ótima do problema de dimensionamento de lotes de um único produto perecível, sem restrições de capacidade e preservação da preparação total. Uma heurística, um procedimento *fix-and-optimize* e uma abordagem por buscas adaptativas em grande vizinhanças são propostas para o problema integrado de planejamento de produção e distribuição. Resultados computacionais em conjuntos de instâncias geradas com base na literatura mostram que os métodos propostos obtiveram performances competitivas com relação a outras abordagens da literatura.

Agradecimentos

A Deus, por ter me guiado através dos problemas de otimização da minha vida. Ele, como grande otimizador que é, sempre me fornece problemas que consigo suportar.

À minha família, pelo amor e suporte. Graças a ela aprendi virtudes importantes, como ter honra, expressar humildade, ser paciente e terno e acima de tudo, ser amigo. À minha mãe, cujo amor sempre me incentivou. Ao meu pai, cuja vida e experiência me enche de inspiração. E à minha irmã, uma companheira dedicada e amorosa.

À minha família aumentada, em especial meus avós Gerolino, Maria Amélia e Anita. Vocês são fontes de ternura e experiência. E sempre me lembro de vocês com lágrimas nos olhos. Aos meus padrinhos Sebastião, Rosa, Luís e Socorro e a todos os meus tios, primos e parentes distantes.

Em especial, à tia Lúcia Helena, sempre presente em minha vida e que nos presenteou com a minha prima mais querida, quase irmã, Hérica. Mal consigo expressar em palavras a saudade imensa de ti e dos seus abraços nada convencionais. Onde quer que esteja, agradeço por ter me iluminado em tantas questões. Amo-te.

À minha orientadora, professora doutora Franklina Maria Bragion de Toledo, cuja paciência e sabedoria são notáveis. Entendo que não sou uma pessoa fácil de lidar, mas o fizeste de uma maneira primorosa.

Ao meu coorientador, o professor doutor Bernardo Sobrinho Simões de Almada Lobo, que por meio de vários conselhos, conversas fraternas e ensinamentos me proporcionou um grande e rico aprendizado, numa terra distante e acolhedora da qual jamais esquecerei.

À professora doutora Maristela Oliveira dos Santos e o professor doutor Cláudio Nogueira de Meneses, que me guiaram através do mestrado e me deram valiosos conselhos.

Ao conjunto de professores que pacientemente me ensinaram diversos conhecimentos começando pela minha infância até aqueles professores que pacientemente me ensinarão no futuro. Espero poder em breve repassar esta sabedoria a mim foi confiada tão bem quanto vocês me passaram. Neste conjunto, ressalto os professores do grupos de otimização do LOT e de Portugal. Espero ter muitos conhecimentos a compartilhar com estas pessoas após ter aprendido tanto.

Ao Laboratório de Otimização (LOT), por disponibilizar conhecimento, amizades e inspiração. Momentos passados no laboratório juntamente com as pessoas que o coabitam me fazem sempre querer estar neste local de trabalho. Em especial, aos amigos Victor Camargo, Marcos Furlan, Gabriela Furtado, Tamara Baldo e Cláudia Fink, Douglas Além e Aline Leão pelos conselhos, ensinamentos e atividades não acadêmicas. Vossa amizade faz sentir-me muito bem.

Ao grupo de Otimização em Portugal, onde passei um ano maravilhoso graças ao

vosso acolhimento e companheirismo. Ao Sam Heshmati e Diana Yomali Ospina pelo carinho, conselhos amigos e pelas aventuras no Porto. Lembro-me de vós com sempre com sorrisos agradecidos. Em especial, ao Pedro Amorim, pelo trabalho conjunto, quase uma co-orientação. Seus conselhos e nossas discussões foram muito importantes para a minha formação científica.

Àqueles presentes nas minhas qualificações e na minha defesa de mestrado, especialmente as bancas, cujas sugestões foram essenciais para o meu trabalho.

À presente banca de doutorado, cujas sugestões, conselhos e correções serão essenciais e engrandecerão este trabalho.

À minha república e agregados, que hoje são a minha atual família de São Carlos. A todos que passaram pela república, um dia, uma semana, um mês ou mais. Carrego comigo toda a fraternidade e alegria contagiante que vocês representam. Em especial, ressalto companheiro inestimáveis, cuja amizade e exemplos me incentivam: Bruno Max, Dário, Maurício, Juari, Márcio André, Berlândia, Brahma, Marcelão e Hugo.

Aos meus amigos e conhecidos de São Carlos, desde a época que comecei, como bixo em engenharia mecatrônica a todos os outros que vim acumulando pelo caminho. Aqui ressalto a minha companheira de aventuras Dani, a minha companheira de risadas bestas Laurenn, a minha companheira da madrugada Aline e minha companheira de assuntos mais filosóficos Marina.

Aos meus amigos que estabeleci em Portugal, das maravilhosas vezes que comemos francesinhas, bebemos vinhos e finos, viajamos, conversamos e rimos. Em especial, a Carlinha por seu jeito brasileiro inconfundível, ao casal mais querido João e Lígia, e às portuguesas Ana Raquel e Sofia. Mais especial ainda, às melhores amigas Ingrid Toth e Marília. Nunca me esquecerei dos nossos surtos psicóticos na madrugada, nossas viagens, conversas e abraços.

A todos meus amigos que deixei em Goiânia quando parti para estudar aqui em São Carlos. Alguns laços se romperam, outros estão mais fortes. Em especial, Brunno Mendes, Sir Fabiano, Rosalinda, Verena, Gabriel, Flávio César, dentre outros tantos.

Às agências de fomento, em especial a FAPESP, sob o processo 2010/06901-1, que fornece a minha bolsa de doutorado e ao CNPq, que me possibilitou fazer o estágio de pesquisa no exterior (processo 208690/2012-3 - Doutorado Sanduíche no Exterior - SWE). Em especial, aos pareceristas destes processos, cujo processo de crivo e apoio da pesquisa é crucial para o desenvolvimento científico nacional.

A todos os funcionários do ICMC, professores, seção de pós graduação, técnicos, guardas e funcionários de limpeza, cujo trabalho tornou a experiência de desenvolver esta tese mais fácil.

Agradeço por último a todos aqueles que não foram citados. Agradeço muito a todos aqueles que participaram de alguma maneira de minha vida. Vocês contribuíram na minha formação social, espiritual, científica e por isso sou muito grato a vocês.

Contents

1	Introduction	1
1.1	Outline of the thesis	5
2	CLSP with setup carryover and crossover	7
2.1	Literature Review	8
2.2	Problem statement and proposed models	9
2.2.1	Literature model	11
2.2.2	First proposed formulation	14
2.2.3	Second proposed formulation	15
2.2.4	Relationship between the proposed models	19
2.2.5	Example	20
2.3	Computational experiments	20
2.3.1	Data generation	21
2.3.2	First test	22
2.3.2.1	Computational Results	23
2.3.3	Second test	25
2.3.3.1	Computational Results	26
2.4	Conclusion	29
3	CLSP with perishable products	31
3.1	Literature Review	32
3.2	Problem statement and proposed models	34
3.2.1	Example	37
3.2.2	Valid Inequalities	39
3.3	Computational experiments	40
3.3.1	Data	40
3.3.2	Computational results	41
3.4	Conclusion	45
4	Lagrangian heuristic for CLSP-PP	47
4.1	Literature review	47
4.2	Problem statement	52
4.3	Lagrangian heuristic	55
4.3.1	Lagrangian relaxation	56
4.3.2	Subgradient optimization	60
4.3.3	Feasibility procedure	61
4.4	Computational study	62
4.5	Conclusion	66
5	Operational integrated production and distribution problem	69

5.1	Literature Review	70
5.2	Problem Statement and Mathematical Formulations	71
5.2.1	Integrated Batch Scheduling and Vehicle Routing Problem (I-BS-VRPTW)	73
5.2.2	Integrated Lot Sizing and Scheduling and Vehicle Routing Problem (I-LS-VRPTW)	76
5.2.3	Relation Between both Models	78
5.3	Computational Study	80
5.3.1	Data Generation	80
5.3.2	Computational Results	83
5.3.3	Solution Examples	86
5.4	Conclusions	89
6	ALNS for the operational integrated production and distribution problem of perishable products	91
6.1	Problem statement	93
6.1.1	Mathematical formulation	93
6.2	Proposed Methods	97
6.2.1	Constructive heuristic	97
6.2.2	Exact Methods	99
6.2.3	Fix-and-Optimize	100
6.2.4	<i>ALNS</i>	102
6.3	Computational experiments	105
6.3.1	Data Generation	105
6.3.2	Computational results	107
6.4	Conclusion	114
7	Conclusion	117
7.1	Perspectives	119
	Bibliography	121
A	Dolan-Moré Chart	131
A.1	Example	131

List of Figures

Figure 2.1 – A solution to the <i>CLSP-BL-SCC</i>	10
Figure 2.2 – Feasible setup variables Z in the proof example.	18
Figure 2.3 – Setup matrix with Z_{15} as a possible setup and the consequent infeasible setups.	18
Figure 2.4 – Solution of the <i>CLSP-BL-SCC</i> example.	20
Figure 2.5 – Average decomposed solution value of <i>Su08</i> as <i>MLST</i> increases for different <i>NILST</i> values.	24
Figure 2.6 – Fraction of the planning horizon capacity loaded with setup and production operations for different <i>NILST</i>	25
Figure 2.7 – Average solution time of <i>Su08</i> versus <i>MLST</i> for different <i>NILST</i>	25
Figure 2.8 – Number of instances with setup crossover (K), <i>RP</i> and <i>SP</i> scenarios: (a) <i>NILST</i> = 1; (b) <i>NILST</i> = 2.	26
Figure 3.1 – Optimal solution to the <i>CLSP-PP</i> example (660 cost units).	38
Figure 3.2 – Optimal solution to the <i>CLSP-PP</i> example relaxing shelf-life constraints (640 cost units).	38
Figure 3.3 – Performance chart for optimality gap.	43
Figure 3.4 – Performance chart for solution gap.	45
Figure 4.1 – <i>DP</i> for problem $LR_i(\lambda, \mu, \nu)$ from period 0 to period T	59
Figure 4.2 – <i>DP</i> for problem $LR_3(\lambda, \mu, \nu)$ from period 0 to period 4.	60
Figure 4.3 – Lagrangean heuristic features over the iterations.	64
Figure 5.1 – Comparing the decision variables of I-BS-VRPTW and I-LS-VRPTW.	79
Figure 5.2 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=4, C-S-TS (St-).	87
Figure 5.3 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=4, C-S-NTS (Seq).	87
Figure 5.4 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=5, P-L-TS (Dist+, St-).	88
Figure 5.5 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=2, C-L-TS (Dist-).	88
Figure 5.6 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=4, P-L-TS (V-, Dist-, St+).	89
Figure 6.1 – Production plan given by the heuristic (<i>Heur</i>).	100
Figure 6.2 – Production plan of the optimal solution.	100
Figure 6.3 – Distribution plan of the optimal solution.	100
Figure 6.4 – Differences between <i>FO_1_0</i> and <i>FO_3_2</i>	102
Figure 6.5 – Performance evaluation of the proposed methods.	112

Figure 6.6 – Performance of the average solution value relative to the warm start
solution in time. 112

Figure 6.7 – Performance of the average solution value relative to the warm start
solution in time, for different instance sizes. 113

Figure A.1 – Performance chart for normalized solution values. 132

List of Tables

Table 2.1 – Number of variables for the <i>CLSP-BL-SCC</i> models.	19
Table 2.2 – Model sizes considering problems with/without long setup times.	20
Table 2.3 – Demand and capacity data.	20
Table 2.4 – Solution values of the <i>CLSP-BL-SCC</i> example.	21
Table 2.5 – Average and maximum relative differences of <i>Kzero</i> solutions in relation to <i>Su08</i> solutions.	23
Table 2.6 – Relative average solution objective value and optimality gap for <i>CLSP- SCC</i>	27
Table 2.7 – Relative average solution objective value and optimality gap for <i>CLSP- BL-SCC</i>	28
Table 3.1 – Remaining data of the example.	38
Table 3.2 – Optimality gaps for <i>CF</i> and <i>FLF</i>	43
Table 3.3 – Average relative difference over solutions for <i>CLSP-PP</i>	44
Table 4.1 – Lagrangean relaxation approaches applied to lot-sizing problems.	53
Table 4.2 – Optimality gap of the compared methods.	65
Table 4.3 – Average relative difference of upper bounds for <i>CLSP-PP</i>	66
Table 4.4 – Average relative difference of lower bounds for <i>CLSP-PP</i>	67
Table 4.5 – Computational times for <i>CLSP-PP</i> (in seconds).	67
Table 5.1 – Gaps between batching and lot-sizing solutions.	84
Table 5.2 – Detailed costs for all instances using the I-BS-VRPTW and I-LS-VRPTW models.	85
Table 6.1 – Demand (dem_{jc}) and Shelf-life (sl_j).	99
Table 6.2 – Travel costs (ct_{cd}) and times (tt_{cd}) and time-windows (a_c, b_c).	99
Table 6.3 – Destroy operators of the <i>ALNS</i>	105
Table 6.4 – Different combinations and the approximate number of binary variables (in thousands).	106
Table 6.5 – Results for the <i>ALNS</i> with different operator time limits.	108
Table 6.6 – Results for the <i>ALNS</i> with different α values.	108
Table 6.7 – Performance evaluation of the operators of the <i>ALNS</i>	109
Table 6.8 – Average solution performance gap and the best optimality gap achieved.	110
Table 6.9 – Average computational times of the best methods.	114
Table A.1 – Absolute and normalized solution value of three approaches.	132

List of Algorithms

Algorithm 4.1	Lagrangean heuristic - LH	56
Algorithm 4.2	Adapted TTM	63
Algorithm 5.1	Pseudo-code to generate production (P) oriented time-windows .	82
Algorithm 5.2	Pseudo-code to generate customer (C) oriented time-windows . .	82
Algorithm 6.1	Constructive heuristic.	98
Algorithm 6.2	Proposed fix-and-optimize heuristic (FO_{x-y}).	101
Algorithm 6.3	Proposed $ALNS$	104
Algorithm 6.4	Pseudo-code to generate time-windows	107

1 Introduction

Production planning refers to the planning of the acquisition of resources and raw materials, as well as the planning of the production activities, required to transform raw materials into finished products meeting customer demand in the most efficient or economical way possible (POCHET; WOLSEY, 2006). The production planning is within the context of supply chain planning, which provides a holistic representation of all company processes, from the supplier to the customer. It involves decisions about the procurement of raw materials, the manufacturing processes and the distribution operations until the sale for the consumer. The proper planning of such activities leads companies to competitive advantages such as: lower production costs; faster, cheaper and reliable deliveries of finished products; more control over the production flow to unexpected events; better customer satisfaction; and many others.

In the context of production planning, companies perform three levels of decisions: strategic, tactical and operational. Strategic planning faces long-term decisions, delineating future directions for the company. Such decisions in production planning denote changes on how the production is performed, for instance, setting up a location to a new plant, or deactivating an unwanted facility or even modifying the production environment. Tactical planning details the “tactics” needed to support the goals envisaged by the strategic planning. This planning performs medium-term decisions such as determining the volume and timing of the finished products to be manufactured in a planning horizon and capacity planning. Operational planning controls the day-to-day decisions in order to achieve the outlined tactical objectives. It consists of short-term decisions such as determining the scheduling of the production orders on the production units and other shop-floor decisions.

Lot sizing is one of the production planning problems concerned with tactical to operational decisions of when to manufacture production orders and the size of these orders. In lot-sizing problems, demand orders are planned as production orders to be processed according to the production environment and the product characteristics. The general objective is the minimization of costs, which are incurred in case of production, setup and holding operations. Depending on the context, other decisions should be integrated, for instance, scheduling, sequencing and resource loading, *i.e.*, the decisions on the instant to initiate and complete the production of a specific item, the sequence of production orders and which resource should be used in that production operation, respectively. Lot-sizing problems are very common in all sorts of industries and the attention received is not surprising, given the importance of inventories in the global economy (GLOCK *et al.*, 2014). Therefore, the literature on lot sizing is massive, with many topics and an increasing trend of publications and reviews. Such reviews are very important to list and classify

the lot-sizing literature and some of them are referred here: De Bodt (1984), Drexl & Kimms (1997), Karimi *et al.* (2003), Brahimi *et al.* (2006a), Zhu & Wilhelm (2006), Jans & Degraeve (2007), Quadt & Kuhn (2007), Jans & Degraeve (2008), Buschkühl *et al.* (2010) and Glock *et al.* (2014).

Lot-sizing problems depend on the features of the production system that should be considered to model the real problem. In their review, Karimi *et al.* (2003) address some of these characteristics related to the planning horizon, product structure and production system. The planning horizon denotes the time interval in which the decision-maker is planning the production activities and assuming the demand. Basically, the planning horizon may be *finite* or *infinite* and modelled continuously or split into discrete time intervals defined as periods. The size of such periods influences the problem modelling. In a planning horizon of many small-sized periods it is likely that each period has one or two production operations. On the contrary, period size may also be designed to fit multiple production operations. Therefore, the size of the period is an important choice and gives rise to the classification of models as *small-bucket* and *big-bucket* problems. The demand may be *dynamic* or *static* if it changes or not over time and *deterministic* or *probabilistic* if it is known or not a priori. Although many lot-sizing problems require that the demand should be met on its due date, in some problems the demand may be satisfied in future periods (*backlogging*) or even unmet (*lost sales*). The problems may be *single-item* or *multi-item*, with the latter case more complex due to the competition of item-related activities on shared resources. Moreover, products may be considered *perishable* and so they can not be held in inventory for a long time, otherwise they spoil. Lot-sizing problems are also classified according to the number of levels of the product structure. The final products may depend only on raw materials (*single level*) or also on intermediary products, which characterises the *multi-level* case. Distinct production shop-floor environments are known in the literature, such as *single and parallel machines*, *flowshop*, *jobshop*, *openshop* and the *flexible* version of the latter three. The most common feature of lot sizing problems is the *capacity* of resources, which limits the production and other related operations, such as the time of the period available for production, manpower and budget. The machines need to be set up for the production of the items, incurring in costs and capacity consumption (mainly times). The setup costs and times may be *constant*, *product-dependent* or be *sequence-dependent*, *i.e.*, to let the machine ready to produce a product, the costs incurred and the time spent is dependent on the predecessor item. Other considered characteristics of setups are *setup carryover* and *setup crossover*. Both mean that the setup state of a machine is maintained from a period to the following one. The former denotes that the machine is ready to process a production order and this machine setup state is carried over to the next period. The latter occurs when the machine is being set up and the setup operation crosses over period boundaries, *i.e.*, the incomplete setup state of the machine is preserved between periods.

All the aforementioned characteristics and many other not referenced here show the broad range of production systems and the specific features/extensions that should be taken into account when modelling a lot-sizing problem. In this thesis two main features are studied in the context of lot-sizing problems: (a) setup crossover; and (b) perishable products.

The setup crossover is an extension of the setup carryover, in which the setup state of a machine ready to produce is carried over between adjacent periods. The setup carryover (also known as linked lot sizes) may avoid one setup operation per period, directly promoting setup cost and time savings and decreasing inventory levels. On the other hand, setup crossover (also known as period-overlapping setup or setup splitting) allow that setup operations may be initiated in one period and be continued to the following one, without any losses between period boundaries. For production planning problem with continuous planning horizons, mathematical formulations that assume discrete time periods and does not assume setup crossover have disadvantages over time continuous models, because solutions of the feasible domain are being neglected. By allowing setup crossovers, flexibility is increased, better solutions can be found and whenever setup times are significant, setup crossovers are needed to assure feasibility (MENEZES *et al.*, 2010). However, few studies have considered setup crossover, due to the inherent complexity of the mathematical formulations.

Therefore, one of the contributions of the thesis is the study of setup crossover assumption on lot-sizing problems. The study includes measuring the impact of such assumption for production systems where some of the products with varying setup times, which may be even larger than a period size. Moreover, the development of novel mixed-integer linear programming mathematical formulations using new modelling approaches for setup variable are analysed. To the best of our knowledge, there is not an instance set for these problems on the literature. Then, a set of instances is proposed and a comparison of the proposed models against a literature model is performed.

Perishable products are present in many industrial supply chains, from procurement to distribution. Perishability is related to the loss of value and the sense of utility of the good. Such loss may be due to spoilage, obsolescence, decay, damage and other processes that deteriorate the good. For production planning problems that deal with perishable products, there is a trade-off between supply chain costs, ageing and freshness of finished products. The concept of perishability depends on the planning horizon considered. In case the shelf-life of the products extends too further the planning horizon, there is no need of assuming this property to modelling of production planning problems. Otherwise, in case the shelf-life of the product is shorter than the planning horizon, then the perishability may be an issue, causing spoiled inventory and related costs. In this context, two ranges of shelf-life were studied: (a) products with medium-term shelf-life; and (b) products highly perishable, with short-term shelf-life. For lot-sizing problems, the former assumption

constrains the problem, though few changes are necessary to tackle perishable products and the planning remains on the tactical level. On the other hand, short shelf-life products requires a more careful control over the production planning and in many cases it even forces the integration with other aspects of the supply chain, for instance the distribution problem.

Lot-sizing problems with medium-term shelf-life have their inventories constrained due to perishability issues. In this case, perishable products with fixed lifetime measured in term of periods are considered. A first-in-first-out policy is used to handle the inventory, *i.e.*, the older products in inventory are sent first to satisfy the demand. For the modelling of this problem, lot size variable reformulation proposed by Krarup & Bilde (1977) provides tighter models, with clear advantages regarding the inventory management. The comparison of this modelling technique against classical models is performed to a set of generated instances.

For products with short-term shelf-life, lot-sizing problems should consider that finished products can not take long to be delivered to customers. This assumption induces the integration of production and distribution planning. Due to the shelf-life, the planning should be taken at an operational level. The literature has usually addressed the operational integrated production and distribution problem without considering lot-sizing/splitting decisions. So, production orders are assumed to be batches of customer demand orders, which makes the problem simpler and it seems that feasible plans have been generated. However, it is a consensus that lot-sizing/splitting decisions are advantageous and sometimes necessary to achieve feasible solutions for operational problems where scheduling decisions are taken jointly. To the best of our knowledge, the incorporation of lot-sizing decisions in the operational production and distribution problem has never been analysed. Therefore, an evaluation on lot-sizing decisions against batching is performed for the operational integrated production and distribution planning problem with perishable products. A secondary contribution discusses the main conditions in which lot sizing may improve production and distribution plans restricted to batching decisions.

The main contributions of the thesis mentioned before are based on the modelling of production planning problems with extensions that deal with real-world features of complex production systems. Mixed-integer linear programming formulations were developed and state-of-the-art optimization software (MILP-solvers) used to solve these problems by means of branch-and-cut procedures. However, MILP-solvers face a broad range of mathematical programming issues, which may constitute a disadvantage against problem-driven solution approaches. Moreover, solution applications are usually limited to a computational time for each problem treated, which does not guarantee the provably optimal solutions for the exact methods of the MILP-solvers. Problem-driven heuristic approaches may deliver better results for the proposed problems. Therefore, another con-

tribution of the thesis relies on the development of simple heuristics, metaheuristics and matheuristics methods for the proposed production planning problems, achieving good-quality results in limited time, mainly for large-size and practical instances.

1.1 Outline of the thesis

The thesis is organized in self-contained chapters, *i.e.*, although the contents of the chapters are connected, each chapter is independently readable and understandable without the contents of the other chapters. The remainder of the thesis is outlined as follows.

Chapter 2 addresses the capacitated lot sizing problem with backlogging and setup carryover and crossover (*CLSP-BL-SCC*). Two novel formulations are proposed and the latter model presents an innovative way to model setup variables, which disaggregates the time index in start and completion time periods of the setup operations. This original idea confers a more compact model in terms of constraints and variables. A thorough study on the impact of setup crossover assumption is conducted, together with an extensive computational comparison of the proposed models against a literature formulation were conducted.

Chapter 3 introduces the capacitated lot sizing problem with setup carryover and perishable products (*CLSP-PP*). Two mixed-integer linear programming models are proposed with a difference regarding the lot sizing variable representation: (a) aggregated, where the variable defines the lot size of an item to be produced in a period; and (b) disaggregated, where the variable denotes the fraction of a demand order to be produced in a period. A comparison of both models is performed using a MILP-solver limited to different computational time limits (1, 10 and 30 minutes).

Chapter 4 provides a lagrangean heuristic approach to address *CLSP-PP*. The lagrangean relaxation of capacity and other time-coupling constraints are considered and the resulting problem is solved by a dynamic programming procedure. The lagrangean dual problem is solved by subgradient optimization and the proposed feasibility procedure is adapted from a well-known method of the literature (TRIGEIRO *et al.*, 1989). Although being a heuristic, this approach allows the measurement of the solution quality through the calculation of a good-quality lower bound. Finally, Chapter 4 performs a comparison of the lagrangean heuristic against the most successful model of Chapter 3.

Chapter 5 defines the operational integrated production and distribution planning problem with perishable items (*OIPDP*). The chapter discusses the importance of considering lot sizing/splitting decisions in this integrated decision environment against the usual batching assumption, *i.e.*, a demand order may be produced in multiple production orders or exclusively by a single batch. The advantages of the lot sizing/splitting assumption are outlined and discussed in detail, showing the impact provided by such assumption. Two novel formulations are proposed, the first considering only batching

decisions and the second performing lot sizing/splitting decisions. The proposed models presented an inherent complexity due to the integration of production and distribution planning decisions and so, are inefficient for practical size problems.

Chapter 6 fulfils this gap, proposing an adaptive large neighbourhood search algorithm (*ALNS*) to tackle *OIPDP*. A simple speed-driven construction heuristic provides an usually low-quality solution, which is used to feed *ALNS*. A data set with large-size instances is generated and computational tests are conducted in order to compare *ALNS* against other known exact and heuristic procedures.

Chapter 7 summarises the contents of the thesis, highlighting the major contributions and proposing perspectives on distinct research areas.

2 CLSP with setup carryover and crossover¹

Setup operations are significant in some production environments and may strongly influence lot-sizing and scheduling decisions. The setup operations prepare the processing unit (machine, line) to manufacture production lots, consuming capacity (denoted by setup times) and incurring setup costs. In some production lines, it is also assumed that the setup state may be fully or partially maintained over periods, denoted in the literature by setup carryover and setup crossover, respectively. The setup carryover and crossover assumptions yield the continuity of scheduling decisions across periods, for production and setup operations, respectively. Such assumptions are appreciated, for instance, by process industries with considerable setup times. Indeed, process industry setups usually deal with extensive cleansing-up operations. Furthermore, testing operations should be performed to guarantee that no contamination affects the downstream processes. Therefore, setup times consume a significant part of the period's length, augmenting the importance of making a flexible assignment and timing of the production and setup operations. Setup carryover and crossover were applied to chemical and beverage industries (SUNG; MARAVELIAS, 2008) and (KOPANOS *et al.*, 2011), respectively.

The setup carryover allows a setup state to be maintained from one period to the next adjacent one. This feature may promote setup cost and time savings and decrease inventory levels. The setup carryover assumption is more common in small-bucket formulations, since setup times may consume a large amount of the micro-period capacity. Once there is at most one setup per period, it is straightforward to consider such a feature. Nevertheless, regarding large-bucket formulations, the literature has assumed the setup carryover due to the cost savings, the more efficient consumption of capacity and the feasibility of instances with tight production capacity.

The setup crossover (also known as period-overlapping setup or setup splitting) defines the opportunity to start a setup operation in one period and continue it to the following one, *i.e.*, the incomplete setup operation crosses over time period boundaries. In case of long setup times (in relation to the size of the period, may be even greater than one period length), the setup operation might be performed in more than two periods. By allowing setup crossovers, flexibility is increased, better solutions can be found and whenever setup times are significant, setup crossovers are needed to assure feasibility (MENEZES *et al.*, 2010). Although setup crossover is a natural extension of the setup carryover, few studies have assumed it, due to the difficulty in dealing with the underlying models. If the planning horizon of the problem is treated as continuous (for instance, 24/7 industrial environments), small-bucket and large-bucket formulations which do not assume

¹ The contents of this chapter are consonant with the paper “Models for capacitated lot-sizing problem with backlogging, setup carryover and crossover”, referenced by (BELO-FILHO *et al.*, 2014).

setup crossover do not take into account all possible solutions of the feasibility domain. Furthermore, without the setup crossover feature, the decision maker is not totally free to choose the period size, which, in this case, would have to be at least the size of the longest setup time.

This chapter details the study outlined in Belo-Filho *et al.* (2014), which approached two novel formulations for the capacitated lot-sizing problem with backlogging and setup carryover and crossover (*CLSP-BL-SCC*). The first formulation applied the setup crossover extension to the capacitated lot-sizing problem with setup carryover (*CLSP-SC*) developed by Suerie & Stadtler (2003). The second formulation institutes a new disaggregated setup variable, which permits an even more compact model. The setup variable disaggregation is inspired on the classical lot-sizing facility location reformulation (KRARUP; BILDE, 1977). The new setup variable is indexed by the periods in which the setup starts and ends, unlike the classical setup variable period index, which indicates when the setup is performed, *i.e.*, the period in which the setup starts. A thorough study on the impact of setup crossover assumption and an extensive computational test including literature and the proposed models were conducted. Computational results show that the proposed models have outperformed other state-of-the-art formulation.

The remainder of the chapter is organised as follows: Section 2.1 provides a brief literature review; Section 2.2 states the problem and presents the literature model along with the two new *CLSP-BL-SCC* formulations; Section 2.3 describes the computational tests and Section 2.4 concludes our study and suggests some directions for further research.

2.1 Literature Review

The capacitated lot-sizing problem with setup carryover and crossover (*CLSP-SCC*) is a relatively new problem and little research has been conducted in this area. Sung & Maravelias (2008) presented a mixed-integer linear programming (*MILP*) large-bucket formulation for the *CLSP-SCC*. It considers non-uniform time periods and long setup times and has been extended to model idle time variations, parallel machines, families of items, backlogging and lost sales. Menezes *et al.* (2010) also formulated the *CLSP-SCC* considering sequence-dependent and non-triangular setups, allowing for sub tours. Kopanos *et al.* (2011) developed a model for *CLSP-BL-SCC* with parallel processing units and items classified into product families. Family changeovers are sequence-dependent, however the setup is sequence-independent for products of the same family. Setup crossover is considered only for family changeover. The model has been extended to tackle processing units that remain idle through an entire period (using a dummy product approach) and maintenance activities. Their approach was applied to the bottling stage of a beer production facility. In Camargo *et al.* (2012), one of the three formulations proposed for the two stage lot-sizing and scheduling problem considers setup crossover, which is achieved by a

continuous-time representation. Mohan *et al.* (2012) extended the *CLSP-SC* formulation of Suerie & Stadtler (2003) to address setup splitting, though the setup operation may be split in at most two periods. For a small set of instances, the author showed that the modelling of setup crossover yielded more feasible solutions and improved solution costs.

In the context of small-bucket formulations, the exact modelling of setup operations is crucial, since the setup times consumes a substantial portion of the length of a period (period's capacity). Cattrysse *et al.* (1993) and Blocher *et al.* (1999) designed formulations based on the discrete lot-sizing and scheduling problem model. However, the setup times were multiple of period's capacity, which constrains the formulation use in practice, since choosing period size becomes more restricted. Drexel & Haase (1995) proposed the proportional lot-sizing and scheduling problem formulation and one extension deals with period overlapping setup times. Although the setup times were considered free to assume any value, Suerie (2006) showed that the formulation proposed by Drexel & Haase (1995) disregard some solutions, by a counter example. Furthermore, Suerie (2006) developed two models for the lot-sizing problem with setup crossover, which outperformed the previous formulations on the quality and flexibility of the solution. Kaczmarczyk (2009) proposed two *MILP* formulations based on the *PLSP* with setup crossover. The results showed a better performance of the new models over the literature, mainly for setup times longer than the period length. In Kaczmarczyk (2013), *PLSP* problem with parallel machines and setup times with period overlapping were studied and one model was presented. The setup operation may be split to at most two periods. A small set of instances were generated and computational tests showed that although computational times were largely increased, a relative averaged decrement of approximately 2% on the total cost was achieved, when setup crossover was assumed.

2.2 Problem statement and proposed models

In the following, we propose two large-bucket alternative models for the *CLSP-BL-SCC* consistent with the problem presented in Sung & Maravelias (2008). The *CLSP-BL-SCC* formulation of Sung & Maravelias (2008) is considered the literature model. The new formulations use other modelling techniques as disaggregation of the binary setup variable, leading to computationally more efficient models. To the best of our knowledge, it is the first model to rely on such a feature.

In the *CLSP-BL-SCC*, the decision maker plans the production lot sizes and scheduling for N products (items) which share a single processing unit (machine, line) over a finite planning horizon composed of T periods. The dynamic and deterministic demand must be met at the end of the planning horizon. Along the horizon, period inventory and backlogging are allowed, incurring costs. Product-dependent setup times and costs are considered. The setup cost is incurred in the period in which the setup operation starts.

The setup state may be preserved across periods, even if the setup operation is not finished. In other words, the setup state may be maintained across adjacent periods regardless the operation being complete (setup carryover) or incomplete (setup crossover). The objective is to minimise the overall cost, which include backlogging, holding and setup.

When the setup crossover is assumed, two new particular production planning scenarios should be recognised. The first scenario occurs when the setup states are the same at the beginning and at the end of the period and other items which require other setup states are produced in the period. This scenario allows the setup state of an item to be active twice in the same period, which is forbidden or cost-prohibitive in the *CLSP-SC* problems *i.e.*, there is a return to the initial product setup state (return product or RP scenario). The second scenario occurs when a setup time is longer than a period width. The setup starts in a period and finishes in one of the following periods. Therefore, an entire period may be dedicated to an in-progress setup operation (setup in progress or SP scenario).

The setup features discussed above are illustrated in the solution example of a Gantt chart (Figure 2.1). Items A, B, C and D are produced within a planning horizon of six non-uniform time periods. The period boundaries are indicated by the vertical lines. Setup times are represented by hatch bars. The white bars denote the production processes. The RP and SP scenarios are illustrated in periods 3 and 5, respectively.

Figure 2.1 – A solution to the *CLSP-BL-SCC*.

Some reformulations of the lot-sizing problem provide tighter *CLSP* models (DENIZEL *et al.*, 2008) and (WU; SHI, 2011). Two reformulations are well known: the simple plant location and the shortest path, proposed by Krarup & Bilde (1977) and Eppen & Martin (1987), respectively. According to Denizel *et al.* (2008) and Wu & Shi (2011), both reformulations yield a similar performance for the *CLSP* with setup times and for the *CLSP-SC*. Without loss of generality, we have chosen the simple plant location reformulation for the proposed models. The literature model has also been reformulated using this approach. The indices, parameters and other variables necessary to the mathematical models are defined in the following.

Indices

i, i'	products (items)
t, t', t''	periods

Parameters

N	number of items, also represent the set of items
T	number of periods, also represent the set of periods
bc_i	backlogging cost of item i per unit per period
hc_i	holding cost of item i per unit per period
sc_i	setup cost for item i
pt_i	processing time of item i per unit
st_i	setup time for item i
cap_t	capacity of line in period t (in time units)
d_{it}	demand for item i in period t
δ	small number

Decision Variables

$X_{itt'}$	fraction of the demand for item i in period t' produced in period t
$Idle_t$	line idle time in period t
$Late_t$	extra time for the setup conclusion in period t
$Late_{it}$	extra time for the setup conclusion for item i in period t
Z_{it}	equals 1 if setup for item i starts in period t (0 otherwise)
$Z_{itt'}$	equals 1 if setup of item i begins in period t and finishes in period t' , for $t' \geq t$ (0 otherwise)
S_{it}	equals 1 if setup state i is active in period t (0 otherwise)
α_{it}	equals 1 if setup state i is active at the beginning of period t (0 otherwise)
β_{it}	equals 1 if setup state i is active at the end of period t (0 otherwise)
K_{it}	equals 1 if setup of item i crosses over the end of period t (0 otherwise)
Y_{it}	equals 1 if period t is in the RP scenario for item i (0 otherwise)
W_t	equals 1 if period t is in the SP scenario (0 otherwise)
Q_t	equals 1 if no setup begins in period t (0 otherwise)

2.2.1 Literature model

The literature model is given by Sung & Maravelias (2008) with the facility location reformulation and will be referred to as *Su08*, that reads:

$$\text{Min} \sum_{i,t,t' < t} bc_i(t' - t)d_{it}X_{itt'} + \sum_{i,t,t' > t} hc_i(t' - t)d_{it}X_{itt'} + \sum_{i,t} sc_iZ_{it}, \quad (2.1)$$

$$\text{s.t.} \sum_t X_{itt'} = 1, \quad \forall i, t' \mid d_{it'} > 0, \quad (2.2)$$

$$Late_{t-1} + \sum_{i,t'} pt_i d_{it'} X_{itt'} + \sum_i st_i Z_{it} + Idle_t = cap_t + Late_t, \quad \forall t, \quad (2.3)$$

$$X_{itt'} \leq S_{it} - K_{it} + Y_{it}, \quad \forall i, t, t', \quad (2.4)$$

$$\sum_{i=1}^N \beta_{it} = 1, \quad \forall t, \quad (2.5)$$

$$\beta_{i,t-1} \leq S_{it}, \quad \forall i, t, \quad (2.6)$$

$$\beta_{it} \leq S_{it}, \quad \forall i, t, \quad (2.7)$$

$$Y_{it} \leq \beta_{i,t-1}, \quad \forall i, t, \quad (2.8)$$

$$Y_{it} \leq \beta_{it}, \quad \forall i, t, \quad (2.9)$$

$$Y_{it} \leq \sum_{i'=1, i' \neq i}^N S_{i't}, \quad \forall i, t, \quad (2.10)$$

$$Y_{it} \geq \beta_{i,t-1} + \beta_{it} + S_{i't} - S_{it} - 1, \quad \forall i, i' \neq i, t, \quad (2.11)$$

$$Z_{it} = S_{it} - \beta_{i,t-1} + Y_{it}, \quad \forall i, t, \quad (2.12)$$

$$Late_t \leq \sum_i (st_i - \delta) K_{it}, \quad \forall t, \quad (2.13)$$

$$K_{it} \leq \beta_{it}, \quad \forall i, t, \quad (2.14)$$

$$Y_{it} \leq K_{it}, \quad \forall i, t, \quad (2.15)$$

$$K_{it} \leq Z_{it}, \quad \forall i, t \mid st_i \leq cap_t, \quad (2.16)$$

$$K_{it} \leq Z_{it} + W_t, \quad \forall i, t \mid st_i > cap_t, \quad (2.17)$$

$$Z_{it} \leq K_{it}, \quad \forall i, t \mid st_i > cap_t, \quad (2.18)$$

$$Z_{it} + W_t \leq 1, \quad \forall i, t \mid \max_i \{st_i\} > cap_t, \quad (2.19)$$

$$W_t \geq \frac{Late_{t-1} - cap_t}{\max_i \{st_i\} - cap_t}, \quad \forall t \mid \max_i \{st_i\} > cap_t, \quad (2.20)$$

$$W_t \leq \frac{Late_{t-1}}{cap_t}, \quad \forall t \mid \max_i \{st_i\} > cap_t, \quad (2.21)$$

$$cap_t - Late_{t-1} + Late_t \leq \left(\max_i \{st_i\} + cap_t \right) (1 - W_t), \quad \forall t \mid \max_i \{st_i\} > cap_t, \quad (2.22)$$

$$X_{itt'}, Z_{it}, Idle_t, Late_t \geq 0, \quad \forall i, t, t', \quad (2.23)$$

$$S_{it}, \beta_{it}, K_{it}, Y_{it}, W_t \in \{0, 1\}, \quad \forall i, t. \quad (2.24)$$

The objective function (2.1) minimises backlogging, holding and setup costs. Constraints (2.2) are inventory balance constraints, which ensure that demand is met at the end of the planning horizon. Capacity constraints (2.3) provide the time balance. As setup crossover is considered, extra time $Late_t$ accounts for the time necessary to finish the setup operation. This time is inherited by the following periods, reducing their available capacity. Due to (2.4), the production of item i in period t is bounded and only occurs if the line is ready for production. Constraints (2.5) determine that a single setup state is preserved at the end of the period. Constraints (2.6) and (2.7) impose that, in case of a setup carryover ($\beta_{it} = 1$), the setup state i occurs in periods t and $t + 1$, respectively. Constraints (2.8) to (2.11) define the *RP* scenario. For the occurrence of the *RP* scenario for item i in period t , the setup state is carried over from period $t - 1$ to t (2.8) and from t to $t + 1$ (2.9). The production of a different item is also required between the two setups of the same item i (2.10). When all these conditions are met, then constraints (2.11) force $Y_{it} = 1$. Equations (2.12) establish the conditions under which setup operation Z_{it} occurs: (i) the setup state i is active in period t although it is not inherited from the previous period ($\beta_{i,t-1} = 0$); and (ii) the setup state i is inherited from the previous period ($\beta_{i,t-1} = 1$) and the *RP* scenario occurs ($Y_{it} = 1$). Constraints (2.13) bound the extra time needed for finishing the setup of item i in period t , implying that this setup operation crosses over the period boundary ($K_{it} = 1$). The setup crossover forces the preservation of the setup state (2.14). The *RP* scenario occurs when the returning state originates from a period overlapping setup (2.15). When a short setup ($st_i \leq cap_t$) crosses over into period $t + 1$, constraints (2.16) impose that the setup starts in period t . Otherwise, when the setup time is longer than the period's capacity, constraints (2.17) force the setup to either start in period t or be in progress all over period t (*SP* scenario). For long setups, any setup starts in a given period and finishes in the following periods, imposing a setup crossover (2.18). Naturally, the occurrence of the setup start and the *SP* scenario are mutually exclusive (2.19). Constraints (2.20) and (2.21) bound the *SP* scenario variable, according to the extra time required in the previous period. Inequalities (2.22) impose the extra time needed for setup in the *SP* scenario. The last two set of constraints (2.23) and (2.24) state

the variable domain. Although Y_{it} could be considered continuous, Sung & Maravelias (2008) concluded that computational times appear to improve when considered binary.

2.2.2 First proposed formulation

The first proposed formulation is built on top of the models of Suerie & Stadtler (2003) and Sung & Maravelias (2008) and intended to be a more compact formulation, eliminating the scenario specific variables Y_{it} and W_t , introduced in Section 2.2.1. This new model is denoted by compact merged literature model (*CMLM*) and is defined as follows:

$$\text{Min } \sum_{i,t,t' < t} bc_i(t' - t)d_{it}X_{itt'} + \sum_{i,t,t' > t} hc_i(t' - t)d_{it}X_{itt'} + \sum_{i,t} sc_i Z_{it}, \quad (2.25)$$

$$\text{s.t. } \sum_t X_{itt'} = 1, \quad \forall i, t' \mid d_{it'} > 0, \quad (2.26)$$

$$\sum_i Late_{i,t-1} + \sum_{i,t'} pt_i d_{it'} X_{itt'} + \sum_i st_i Z_{it} \leq cap_t + \sum_i Late_{it}, \quad \forall t, \quad (2.27)$$

$$X_{itt'} \leq Z_{it} - K_{it} + \alpha_{it}, \quad \forall i, t, t', \quad (2.28)$$

$$\sum_i \alpha_{it} = 1, \quad \forall t, \quad (2.29)$$

$$\alpha_{it} \leq Z_{i,t-1} + \alpha_{i,t-1}, \quad \forall i, t, \quad (2.30)$$

$$\alpha_{i,t+1} + \alpha_{it} \leq 1 + Q_t + Z_{it}, \quad \forall i, t, \quad (2.31)$$

$$Z_{it} + Q_t \leq 1, \quad \forall i, t, \quad (2.32)$$

$$Late_{it} \leq (st_i - \delta)K_{it}, \quad \forall i, t, \quad (2.33)$$

$$K_{it} \leq \alpha_{i,t+1}, \quad \forall i, t, \quad (2.34)$$

$$K_{it} \leq Z_{it}, \quad \forall i, t \mid st_i \leq cap_t, \quad (2.35)$$

$$K_{it} \leq Z_{it} + \frac{Late_{i,t-1}}{cap_t}, \quad \forall i, t \mid st_i > cap_t, \quad (2.36)$$

$$Z_{it} \leq K_{it}, \quad \forall i, t \mid st_i > cap_t, \quad (2.37)$$

$$cap_t - Late_{i,t-1} + Late_{it} \leq (st_i + cap_t)(3 - K_{i,t-1} - K_{it} - Q_t), \quad \forall i, t \mid st_i > cap_t, \quad (2.38)$$

$$Q_t \geq \frac{Late_{i,t-1} - cap_t}{st_i - cap_t}, \quad \forall i, t \mid st_i > cap_t, \quad (2.39)$$

$$X_{itt'}, Late_{it} \geq 0, \quad \forall i, t, t', \quad (2.40)$$

$$Z_{it}, \alpha_{it}, K_{it}, Q_t \in \{0, 1\}, \quad \forall i, t. \quad (2.41)$$

The objective function (2.25) minimises the sum of backlogging, holding and setup costs. Equations (2.26) ensure that the demand is met at the end of the planning horizon. Capacity constraints (2.27) limit production and setup operations, considering the time delayed for the following periods in case of setup crossover. Production may occur only if the line is appropriately set up (2.28). The occurrence of this condition is twofold: (i) there is a setup which starts in the current period and does not cross over; and (ii) the setup state is inherited from the previous period. At most one single setup state is preserved between two periods (2.29). Constraints (2.30) indicate the origin of the setup carryover of period t (from either the previous period setup carryover or a setup starting in period t). Inequalities (2.31) determine that a consecutive setup carryover of the same item requires either a period without any setup or another setup operation of the same item. The no setup scenario in period t , *i.e.*, $Q_t = 1$, is defined by (2.32). The extra time needed for a setup crossover is limited in (2.33). In (2.34), a setup crossover in period t implies that the setup state is carried over from period t to period $t + 1$. For short setups, the setup crossover is dependent upon the occurrence of the setup (2.35). For long setups, constraints (2.36) ensure that the setup crossover exists if the corresponding setup operation begins in the period or a setup operation is in progress across the period. The setup in progress denotes that the extra time needed for the previous period is longer than the capacity of the period ($Late_{i,t-1} > cap_t$). Constraints (2.37) ensure that a setup crossover always occurs for items with long setups. Inequalities (2.38) guarantee the proper counting of the extra time needed when a period is in the SP scenario, *i.e.*, when both period boundaries are crossed over by the same setup operation and no setup starts in this period. Constraints (2.39) ensure that no setup starts in a period with an SP scenario. The last constraints (2.40) and (2.41) state the domain of the variables.

2.2.3 Second proposed formulation

In all lot-sizing models reported in the literature (as well as the models of Sections 2.2.1 and 2.2.2), the setup related variables (Z_{it} and S_{it}) are solely indexed by the time period in which the setup operation occurs/begins. This study is the first to propose

a disaggregation of the time index, clearly defining the start and the completion time periods of the setup operation. The new setup variable $Z_{itt'}$ tracks the start (t) and end (t') time periods of the setup operation of item i . The second formulation (Disaggregated Setup Model - *DSM*) is based on this new variable. Therefore, the variables denoted for setup crossover, RP and SP scenarios can be neglected, in comparison to the models *Su08* and *CMLM*. The *DSM* is defined as follows:

$$\text{Min } \sum_{i,t,t' < t} bc_i(t' - t)d_{it}X_{itt'} + \sum_{i,t,t' > t} hc_i(t' - t)d_{it}X_{itt'} + \sum_{i,t,t' \geq t} sc_iZ_{itt'} \quad (2.42)$$

$$\text{s.t. } \sum_t X_{itt'} = 1, \quad \forall i, t' \mid d_{it'} > 0, \quad (2.43)$$

$$\sum_i Late_{i,t-1} + \sum_{i,t'} pt_id_{it'}X_{itt'} + \sum_{i,t' \geq t} st_iZ_{itt'} \leq cap_t + \sum_i Late_{it}, \quad \forall t, \quad (2.44)$$

$$X_{itt'} \leq \alpha_{it} + Z_{itt} - \sum_{t'' < t, t''' > t} Z_{itt''t'''}, \quad \forall i, t, t', \quad (2.45)$$

$$\sum_i \alpha_{it} = 1, \quad \forall t, \quad (2.46)$$

$$\alpha_{i,t+1} \leq \sum_{t' \geq t} Z_{itt'} + \alpha_{it}, \quad \forall i, t, \quad (2.47)$$

$$\alpha_{i,t+1} + \alpha_{it} \leq 1 + Q_t + \sum_{t' > t} Z_{itt'}, \quad \forall i, t, \quad (2.48)$$

$$\sum_{t' \geq t} Z_{itt'} + Q_t \leq 1, \quad \forall i, t, \quad (2.49)$$

$$\sum_{i,t' > t} Z_{itt'} \leq 1, \quad \forall t, \quad (2.50)$$

$$\sum_{i,t' < t} Z_{it't} \leq 1, \quad \forall t, \quad (2.51)$$

$$\sum_{t' < t, t'' \geq t} Z_{it't''} \leq \alpha_{it}, \quad \forall i, t, \quad (2.52)$$

$$Late_{it} \leq \sum_{t' \leq t, t'' > t} (st_i - \delta)Z_{it't''}, \quad \forall i, t, \quad (2.53)$$

$$cap_t - Late_{i,t-1} + Late_{it} \leq (st_i + cap_t) \left(1 - \sum_{i,t' < t, t'' > t} Z_{it't''} \right), \quad \forall i, t \mid st_i > cap_t, \quad (2.54)$$

$$\sum_{i,t' < t, t'' > t} Z_{it't''} \leq Q_t, \quad \forall t \mid \max_i st_i > cap_t, \quad (2.55)$$

$$Q_t \geq \frac{Late_{i,t-1} - cap_t}{st_i - cap_t}, \quad \forall i, t \mid st_i > cap_t, \quad (2.56)$$

$$\sum_{i,t' < t, t'' > t} Z_{itt'} \leq \frac{Late_{i,t-1}}{cap_t}, \quad \forall i, t \mid st_i > cap_t, \quad (2.57)$$

$$X_{itt'}, Late_{it} \geq 0, \quad \forall i, t, t', \quad (2.58)$$

$$\alpha_{it}, Z_{itt'}, Q_t \in \{0, 1\}, \quad \forall i, t, t'. \quad (2.59)$$

The objective function (2.42) minimises the sum of backlogging, holding and setup costs. The setup costs are incurred in the period in which the setup operation starts. The demand satisfaction and capacity constraints are given by (2.43) and (2.44), respectively. Production in period t is allowed only if either the line is set up at the beginning of the period or a setup started and completed in this period occurs (2.45). Setup carryover is mutually exclusive for items per period (2.46). Constraints (2.47) indicate the origin of the setup carryover of period t (from either the previous period setup carryover or a setup starting in period t). According to constraints (2.48), a consecutive setup carryover is permitted if no setup occurs or a setup crossover is performed. If there is a setup in a period, then $Q_t = 0$, according to (2.49). Constraints (2.50) and (2.51) ensure that at most one setup crossover starts and ends in each period, respectively. Setup crossover implies the preservation of the setup state (setup carryover) by (2.52). Constraints (2.53) and (2.54) determine the extra time due to the setup crossover, even for a period under the SP scenario. Inequalities (2.55) and (2.56) impose that $Q_t = 1$ for the SP scenario period. Constraints (2.57) force the extra time required in the previous period to be longer than the period length in case the period is in the SP scenario. We can observe that (2.56) and (2.57) are not necessary to define the problem properly; they were added as valid inequalities to facilitate the comparison of the models proposed in this chapter. Domain constraints are given by (2.58) and (2.59).

According to the definition, there are $\frac{NT(T+1)}{2}$ binary variables $Z_{itt'}$ (all that respect $t' \geq t$). However, due to the setup times and the period length, only some of these variables represent, in fact, feasible setups. The infeasible setup variables should be discarded for the sake of computational performance improvement. The next proposition discusses this issue.

Proposition 2.1. *There are at most $2NT - N$ binary variables $Z_{itt'}$ which represent feasible setups for the CLSP-BL-SCC.*

Proof. We prove the statement by showing that some setup variables are mutually exclusive. The number of variables $Z_{itt'}$ allocated to item i is independent of the other items.

Setup $Z_{itt'}$ is possible if and only if (1) $st_i \leq \sum_{s=t}^{t'} cap_s$ and (2) $st_i > \sum_{s=t+1}^{t'-1} cap_s$ (naturally, in case $t' - 1 < t + 1$, the sum is zero). Condition (1) indicates whether $Z_{itt'}$ setup time fits the cumulated length from periods t to t' . Condition (2) expresses that in order to turn one $Z_{itt'}$, the respective setup time has to be longer than the sum of respective periods in the SP scenario. Therefore, if $Z_{it,t+2}$ is feasible, *i.e.*, the setup operation starts in period t and ends in period $t + 2$, then clearly the setup time should be longer than period $t + 1$ length. For instance, consider a single item problem and a planning horizon with 5 periods of capacity 12, 2, 2, 2 and 8 time units, respectively. Let the product setup time be 8 time units. Figure 2.2 shows some potential setups for this instance, given by variables $Z_{tt'}$ (single item problem). In the example, Z_{15} is feasible, because conditions (1) $st \leq \sum_{s=1}^5 cap_s$ and (2) $st > \sum_{s=2}^4 cap_s$ hold. Condition (2) implies that periods 2 to 4 are in the SP scenario. However, as condition (2) holds for Z_{15} , then condition (1) for variables Z_{22} , Z_{23} , Z_{24} , Z_{33} , Z_{34} and Z_{44} is not satisfied, which implies that these variables are infeasible. Figure 2.3 shows the setup matrix with all variables $Z_{tt'}$. The highlighted variable Z_{15} is feasible, therefore struck out variables are infeasible.

Figure 2.2 – Feasible setup variables Z in the proof example.

$$\begin{pmatrix} Z_{11} & Z_{12} & Z_{13} & Z_{14} & \textcircled{Z_{15}} \\ & \text{Z}_{22} & \text{Z}_{23} & \text{Z}_{24} & Z_{25} \\ & & \text{Z}_{33} & \text{Z}_{34} & Z_{35} \\ & & & \text{Z}_{44} & Z_{45} \\ & & & & Z_{55} \end{pmatrix}$$

Figure 2.3 – Setup matrix with Z_{15} as a possible setup and the consequent infeasible setups.

Generalizing, in case $Z_{itt'}$ is feasible for $t' \geq t + 2$, through condition (2) $st_i > \sum_{s=t+1}^{t'-1} cap_s$, which means that all variables $Z_{iss'}$, $\forall t + 1 \leq s \leq s' \leq t' - 1$ are infeasible, due to condition (1). In particular, for all anti-diagonals of the setup matrix at most one element is feasible, *i.e.*, at most one element of each anti-diagonal can have conditions (1) and (2) satisfied. As a square matrix of size T has only $2T - 1$ counter-diagonals, there are at most $2T - 1$ feasible setups. For instance, when all setup times are shorter than periods length, the feasible setup variables correspond to the upper bidiagonal matrix. So, there are $2T - 1$ possible setups for each item, or $2NT - N$ binary

variables $Z_{itt'}$ for all the items. In case the setup time for item i is longer than the last period capacity ($st_i > cap_T$), there are fewer than $2T - 1$ feasible setups. Therefore, at most $2NT - N$ binary variables $Z_{itt'}$ may potentially turn on one. $\square$

2.2.4 Relationship between the proposed models

Let $P_{CMLM-LP}$ and P_{DSM-LP} denote the feasible sets of the linear relaxations of formulations *CMLM* and *DSM*, respectively. In the following theorem, we show that *DSM* is at least as strong as *CMLM*.

Theorem 2.1. *The DSM provides a lower bound (using linear relaxation) at least as strong as the CMLM lower bound. In other words, $P_{DSM-LP} \subseteq P_{CMLM-LP}$.*

Proof. To prove the theorem, we first state the equivalence of some variables of *CMLM* and *DSM*. Variables $Late_{it}$, α_{it} , $X_{itt'}$ and Q_t hold the same definition in both models. The relation between the remaining variables Z_{it} and K_{it} of the *CMLM* and $Z_{itt'}$ of the *DSM* is expressed by equations (2.60) and (2.61).

$$Z_{it} = \sum_{t' \geq t} Z_{itt'} \quad \forall i, t \quad (2.60)$$

$$K_{it} = \sum_{t' \leq t, t'' > t} Z_{itt't''} \quad \forall i, t \quad (2.61)$$

Taking into account (2.60) and (2.61), it is easy to see that all the constraints of the *CMLM* are equivalent to those of the *DSM*. Some of these constraints are even fortified. For instance, although Constraints (2.31) and (2.48) are equivalent, the latter does not consider the part which represents the setup starting and finishing in the same period (Z_{itt}). Constraints (2.35) and (2.37) are direct consequences of the definition of the disaggregated variable $Z_{itt'}$. The *DSM* also has some additive constraints, namely (2.50), (2.51) and (2.55), clearly showing that $z_{CMLM-LP} \leq z_{DSM-LP}$. $\square$

A comparison of the sizes of the models has been performed. Table 2.1 quantifies the number of real and integer variables of the models. The two proposed models use fewer integer variables. Table 2.2 quantifies the number of linear constraints of the models, considering the formulations for problems with zero and α ($1 \leq \alpha \leq N$) long setup times. Among the models, *DSM* is the most compact formulation, closely followed by *CMLM*.

Table 2.1 – Number of variables for the *CLSP-BL-SCC* models.

	Real	Integer
<i>Su08</i>	$NT^2 + NT + 2T$	$4NT + T$
<i>CMLM</i>	$NT^2 + NT$	$3NT + T$
<i>DSM</i>	$NT^2 + NT$	$3NT - N + T$

Table 2.2 – Model sizes considering problems with/without long setup times.

	short setups	α long setups
<i>Su08</i>	$N^2T + NT^2 + 8NT + 3T$	$N^2T + NT^2 + 9NT + (\alpha + 6)T$
<i>CMLM</i>	$NT^2 + 7NT + 2T$	$NT^2 + 7NT + (3\alpha + 2)T$
<i>DSM</i>	$NT^2 + 6NT + 4T$	$NT^2 + 6NT + (3\alpha + 5)T$

2.2.5 Example

Consider the following example with 4 items and a planning horizon composed of 6 non uniform periods. Consider product processing times $pt_i = 0.1$ and $hc_i = st_i = sc_i = \{3.0, 4.0, 1.0, 9.0\}$. Table 2.3 shows the remaining data.

Models *Su08*, *CMLM* and *DSM* are employed to solve this instance to optimality. The models provided the same optimal solution illustrated in Figure 2.4. The positive variables of the formulations are shown in Table 2.4. This solution has neither inventory or backlog. The RP scenario takes place in period 3 for the setup state of item B, whereas the SP scenario occurs in period 5 while the machine is being set up for product D. Notice that *DSM* uses fewer variables to represent a solution than the other formulations.

Table 2.3 – Demand and capacity data.

d_{it}	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$
$i = A$	0	30	0	0	0	0
$i = B$	40	20	20	20	0	0
$i = C$	0	0	20	0	0	0
$i = D$	0	0	0	0	0	40
cap_t	10	10	6	6	6	6

Figure 2.4 – Solution of the *CLSP-BL-SCC* example.

2.3 Computational experiments

This section describes two sets of computational experiments. The first investigates the influence of the setup time size in relation to the period length for the *CLSP-BL-SCC*. The second test allows a performance comparison of a state-of-the-art literature model and the two proposed models. An instance data generator was developed considering several different parameters.

Table 2.4 – Solution values of the *CLSP-BL-SCC* example.

<i>Su08</i> variables					
$X_{A22} = 30$	$X_{B11} = 40$	$X_{B22} = 20$	$X_{B33} = 20$	$X_{B44} = 20$	$X_{C33} = 20$
$X_{D66} = 40$	$Late_1 = 1$	$Late_3 = 3$	$Late_4 = 8$	$Late_5 = 2$	
$Z_{A1} = 1$	$Z_{B1} = 1$	$Z_{B2} = 1$	$Z_{B3} = 1$	$Z_{C3} = 1$	$Z_{D4} = 1$
$S_{A1} = 1$	$S_{A2} = 1$	$S_{B1} = 1$	$S_{B2} = 1$	$S_{B3} = 1$	$S_{B4} = 1$
$S_{C3} = 1$	$S_{D4} = 1$	$S_{D5} = 1$	$S_{D6} = 1$		
$\beta_{A1} = 1$	$\beta_{B2} = 1$	$\beta_{B3} = 1$	$\beta_{D4} = 1$	$\beta_{D5} = 1$	$\beta_{D6} = 1$
$K_{A1} = 1$	$K_{B3} = 1$	$K_{D4} = 1$	$K_{D5} = 1$	$Y_{B3} = 1$	$W_5 = 1$
CMLM variables					
$X_{A22} = 30$	$X_{B11} = 40$	$X_{B22} = 20$	$X_{B33} = 20$	$X_{B44} = 20$	$X_{C33} = 20$
$X_{D66} = 40$	$Late_{A1} = 1$	$Late_{B3} = 3$	$Late_{D4} = 8$	$Late_{D5} = 2$	
$Z_{A1} = 1$	$Z_{B1} = 1$	$Z_{B2} = 1$	$Z_{B3} = 1$	$Z_{C3} = 1$	$Z_{D4} = 1$
$\alpha_{A2} = 1$	$\alpha_{B3} = 1$	$\alpha_{B4} = 1$	$\alpha_{D5} = 1$	$\alpha_{D6} = 1$	$Q_5 = 1$
$K_{A1} = 1$	$K_{B3} = 1$	$K_{D4} = 1$	$K_{D5} = 1$		
DSM variables					
$X_{A22} = 30$	$X_{B11} = 40$	$X_{B22} = 20$	$X_{B33} = 20$	$X_{B44} = 20$	$X_{C33} = 20$
$X_{D66} = 40$	$Late_{A1} = 1$	$Late_{B3} = 3$	$Late_{D4} = 8$	$Late_{D5} = 2$	
$Z_{A12} = 1$	$Z_{B11} = 1$	$Z_{B22} = 1$	$Z_{B34} = 1$	$Z_{C33} = 1$	$Z_{D46} = 1$
$\alpha_{A2} = 1$	$\alpha_{B3} = 1$	$\alpha_{B4} = 1$	$\alpha_{D5} = 1$	$\alpha_{D6} = 1$	$Q_5 = 1$

2.3.1 Data generation

The instance generator is inspired by the instance generators available in the literature for problems without the extension of setup crossover. However, to the best of our knowledge, there are no benchmarks in the literature which consider setup crossover, backlogging and long setup times.

For a random number generation (considering a uniform distribution), we used an implementation of the multiplicative linear congruential generator (PARK; MILLER, 1988), with parameters 16,807 (multiplier) and $2^{31} - 1$ (prime number). This algorithm chooses an integer number of the closed integer interval $[a, b]$, a, b integers (hereby denoted as $U[a, b]$).

The instance generator was developed based on problems with backlogging and several parameters were stated. The first parameters considered are number of items (N) and number of periods (T) of the planning horizon. All periods have a fixed length of 1000 time units. The processing times are set to 1 time unit per product unit. The setups are classified as short and long, according to the setup time consumption of the period capacity. The short setup times are generated by $100 + 25 * U[-2, 2]$, *i.e.*, they vary between 5% and 15% of the period length. The *number of items with long setup time*

is defined by the *NILST* parameter. For long setup times, the *mean long setup time* (*MLST*) parameter indicates the size of the setup time in relation to the period length. The long setup times are generated by $MLST + 50\kappa * U[-2, 2]$. Parameter κ denotes the option to vary the *MLST*, $\kappa \in \{0; 1\}$. For example, in case $MLST = 700$ and $\kappa = 1$, long setup times consume a random multiple of 50 from 600 to 800 time units, *i.e.*, 60%, 65%, 70%, 75% or 80% of the period capacity.

In order to draw the demand, we have defined a priori the capacity utilisation needed to process all the demand to 60% of the total planning horizon capacity. The demand is then generated by randomly adding orders of size $50 + 5 * U[-4, 4]$, until the fulfilment of the capacity utilisation. The setup costs correspond to the setup times ($sc_i = st_i$). To calculate the holding costs we rely on the same integer parameter used by Trigeiro *et al.* (1989), the time-between-orders (*TBO*), given by $U[1, 4]$ (average demand of 50 units). The backlogging costs are $U[2, 6]$ times higher than the holding costs.

To summarise, the parameters of the instance generator are N , T , *NILST*, *MLST* and κ . The former two parameters define the instance size and the latter three are related to setup settings. The two tests utilise distinct parameters, which will be detailed in their respective sections.

2.3.2 First test

The first test measures the impact of setup crossover, considering products with long setup times for the *CLSP-BL-SCC*. Small-sized instances were used for this first set of computational experiments. The number of products and periods are fixed to $N = 3$ and $T = 20$. To assess the influence of long setup times on solutions, the setup times should be diversified. The number of items with long setup times (*NILST*) is set to 1 or 2 items. To define the size of the long setup times, two steps are needed. The first step is to state base instances, which maintain the same data for the instances, varying only the size of the long setup time. The size of the long setup times for the base instances is then adjusted to $MLST = 200$ and $\kappa = 0$ (20% of period length). Note that for the base instances the short and long setup times have the same order of magnitude. For each *NILST*, 100 random instances were generated, totalizing 200 instances. The second step relies on increasing the long setup times of the base instances by 100 time units until the long setup times have reached 2500 time units. In other words, the long setup time varies from 0.2 to 2.5 times the period length, considering long setup times of 200, 300, 400 time units and so on (analogously, *MLST* changes its value). Therefore, for each base instance type, 24 instances are generated according to the second step, which provides a total of 4800 instances.

For this test, we take into account a model that does not consider setup crossover and another that assumes setup crossover, RP and SP scenarios. For the case with no crossover, we rely on our first formulation without crossover variables (*i.e.*, $K_{it} = 0$ for

every i and t). Hereafter, this model is referred to as *Kzero*. For the other case, we have adopted the literature model *Su08*.

2.3.2.1 Computational Results

All computational experiments were performed on an Intel Core i5 processor, with 2.80 GHz CPU and 8GB RAM under Linux Ubuntu 10.04 (64 bit). CPLEX version 12.2 from IBM was used as the MIP solver. The data generator described above was used to obtain the instance set. The computational time to solve each MIP was limited to 600 seconds and the parallel mode was active (4 cores).

All instances were tested by *Su08* and *Kzero*. *Kzero* does not solve instances whose setup times are longer than a period ($MLST > 1000$), because the setup crossover is essential for these instances. Table 2.5 shows the average and maximum relative increase in costs when setup crossover is not permitted. In parentheses is the number of instances with increased costs. All *Su08* solutions are better than or equal to *Kzero* solutions. The results show that the extra cost incurred is significant, as the long setup times increase. The number of instances improved by allowing setup crossover is also proportional to the long setup times.

Table 2.5 – Average and maximum relative differences of *Kzero* solutions in relation to *Su08* solutions.

$\frac{Kzero-Su08}{Su08}$ $MLST$	$NILST = 1$		$NILST = 2$	
	Average	Max	Average	Max
200	0.00% (1)	0.33%	0.01% (2)	0.46%
300	0.07% (6)	2.86%	0.14% (19)	2.12%
400	0.04% (14)	0.64%	0.78% (56)	6.33%
500	0.46% (29)	7.63%	1.60% (74)	7.65%
600	1.41% (55)	18.08%	3.87% (96)	15.18%
700	4.17% (86)	22.70%	7.82% (100)	23.90%
800	11.84% (100)	28.84%	14.88% (100)	31.34%
900	26.54% (100)	68.20%	21.67% (100)	34.81%
1000	44.92% (100)	99.11%	27.20% (100)	44.98%

Figure 2.5 illustrates the profiles of the solution value for *Su08*, $NILST=1$ and $NILST=2$ as the setup times increase. The bars indicate the average absolute solution value shared by the costs components. These components represent the costs incurred by the inventory levels, backlog and setup operations, which are denoted by black, grey and white bars, respectively. The cost profiles for *Kzero* and *Su08* are similar. The differences are related to $NILST$, which proportionally affects the rate of increase of the solution cost. The increase in the inventory and backloging costs is evident as $MLST$ increases, reducing the available capacity. The number of setups is reduced and the setup costs decrease.

Figure 2.5 – Average decomposed solution value of *Su08* as *MLST* increases for different *NILST* values.

Figures 2.6 and 2.7 illustrate the impact of setup times on the capacity utilisation and computational times profiles, respectively, for *Su08*, *NILST*= 1 and *NILST*= 2. The idle capacity decreases due to the longer setup times until a very tight configuration. Then, the capacity utilisation becomes unstable, decreasing and increasing as the number of long setup operations is naturally reduced. As a consequence of this reduction, the capacity is freed, however inventory and backlogging costs are incurred. This phenomenon occurs more intensively for *NILST*= 2. The computational times of the solution behave differently. On average, the worst time performance of the solver is around *MLST* equal to 700 and 800 units, for both *NILST*s. For $MLST \leq 1000$, *Su08* spends 55% and 122.8% more computational time than *Kzero*, for *NILST* equal to 1 and 2, respectively. When the capacity utilisation becomes oscillatory for *NILST*= 2, the time performance of the algorithm returns to the same magnitude of the computational times of the smallest *MLST*s, *i.e.*, longer *MLST*s and greater *NILST* made the problem tighter and with less solutions, shorten the computational times on the solver.

A last analysis aims at visualising the frequency of the setup crossover and the consequent *RP* and *SP* scenarios. Figure 2.8 shows the number of instances in which these events occur for *NILST*= 1 and *NILST*= 2. The number of instances with crossover operations (*K*), *SP* and *RP* scenarios is represented by white, grey and black bars, re-

Figure 2.6 – Fraction of the planning horizon capacity loaded with setup and production operations for different $NILST$.

Figure 2.7 – Average solution time of *Su08* versus $MLST$ for different $NILST$.

spectively. For instance, for $MLST = 800$ and $NILST = 1$, 33 out of 100 instances show the *RP* scenario. The average number of setup crossover operations per instance type is provided in the top axis of the chart. Both setup crossover and *SP* scenario become imperative for an $MLST$ larger than 1000 and 2000, respectively. However, all instances show setup crossover operations and *SP* scenario periods for $MLST$ larger than 800 and 1800, respectively for $NILST = 1$. The chart profile for $NILST = 2$ is analogous. Naturally, $NILST$ intensifies the number of setup crossover operations and *SP* scenario periods. However, the *RP* scenario occurs more often for a small $NILST$.

2.3.3 Second test

The second set of tests is based on the first test, except that now instances for both *CLSP-SCC* and *CLSP-BL-SCC* are generated. The objective is to compare the three models available for the problem.

The number of products (N) varies among 5, 10 and 15 items. The planning horizon size (T) varies between 20, 30 and 40 uniform periods. $NILST$ is fixed to 20% of the number of products. The short setup times are chosen by $50 + 10 * U[-2, 2]$. Based on the first test, the *mean long setup time* ($MLST$) is set to 400, 700 and 1200 time units per setup operation and $\kappa = 1$. The first $MLST$ value is more aligned to the lot-sizing literature. Value 700 was chosen due to the time performance of the solver in the first

Figure 2.8 – Number of instances with setup crossover (K), RP and SP scenarios: (a) $NILST = 1$; (b) $NILST = 2$.

test (Figure 2.7). The last $MLST$ value is higher than the period size, therefore the SP scenario may occur.

Considering all the parameters used, we obtained $|N| * |T| * |MLST| = 27$ different combinations. For each combination, 10 random instances were generated, totalizing 270 instances for formulations with backlogging.

For the sake of feasibility for the $CLSP-SCC$ (no backlogging), some periods were inserted at the beginning of the planning horizon, with null demand. The number of inserted periods depends on the combination of parameters N , T and $MLST$, which is given by $\frac{N}{5} + \frac{T}{10} + \left\lceil \frac{MLST}{400} \right\rceil - 2$. For example, an instance of $CLSP-BL-SCC$ with 5 items, 30 periods and $MLST$ of 1200 has 5 extra periods at the beginning of the planning horizon for $CLSP-SCC$.

2.3.3.1 Computational Results

The computational experiments were performed on the same hardware and software previously mentioned. Again, the computational time to solve each MIP was limited to 600 seconds and the parallel mode was active (4 cores).

The proposed models $CMLM$ and DSM were compared to the formulation of Sung &

Maravelias (2008). This test was separated in two parts. Backlogging was disregarded in the former (*CLSP-SCC*) and considered in the latter (*CLSP-BL-SCC*).

Tables 2.6 and 2.7 show the relative average solution objective function value of the three models for *CLSP-SCC* and *CLSP-BL-SCC*, respectively. In parentheses are the optimality gaps of the three models. The columns represent the results of each model. The comparison is focused on (1) the relative difference between the incumbent solution of each model and the best solution achieved by the three models and (2) the optimality gap, *i.e.*, the final relative difference between the incumbent solution and the lower bound. The first rows represent the maximum and the average relative solutions and optimality gaps (in parentheses). The next two lines provide lower bound information. The first line reports the relative average lower bound achieved by the linear relaxation, *i.e.*, the average of the lower bound of each method relative to the best lower bound found by the three models. Analogously, the second line refers to the relative average lower bound achieved at the end of the run. The following lines show the number of instances not solved and solved to optimality by each model, respectively. The relative solution and optimality gaps (in parentheses) are shown for each type of instances.

Table 2.6 – Relative average solution objective value and optimality gap for *CLSP-SCC*.

	DSM		CMLM		Su08	
Maximum	15.21%	(59.55%)	23.87%	(57.77%)	58.40%	(61.36%)
Average	0.71%	(15.25%)	0.97%	(14.85%)	2.93%	(16.25%)
Average Lower Bound						
Linear relaxation	10.72%		12.04		0.00%	
After run	15.08%		14.71%		17.00%	
No solution	3		4		20	
Optimal solution	46		46		45	
Items						
5	0.40%	(7.92%)	0.46%	(7.95%)	0.38%	(9.12%)
10	0.86%	(14.74%)	0.95%	(14.73%)	3.42%	(17.7%)
15	0.88%	(23.37%)	1.54%	(22.19%)	5.50%	(23.31%)
Periods						
20	0.18%	(4.63%)	0.30%	(4.49%)	1.69%	(8.41%)
30	0.64%	(17.3%)	1.51%	(17.01%)	3.55%	(18.67%)
40	1.33%	(24.13%)	1.12%	(23.43%)	3.72%	(22.8%)
Long Setup						
400	0.13%	(3.96%)	0.18%	(3.78%)	0.44%	(4.76%)
700	0.56%	(16.73%)	0.36%	(16.05%)	3.20%	(19.91%)
1200	1.47%	(25.4%)	2.45%	(25.18%)	5.80%	(26.32%)

According to Table 2.6, the *DSM* achieved the best solutions on average within the limited time. However, the best optimality gaps belong to the *CMLM*, due to its better lower bound performance at the end of the run. The number of instances in which no solution was found is considerably higher for *Su08* in comparison with the proposed models. This result shows that the new models are much more robust than the current

Table 2.7 – Relative average solution objective value and optimality gap for *CLSP-BL-SCC*.

	DSM		CMLM		Su08	
Maximum	21.72%	(65.38%)	41.66%	(74.93%)	604.11%	(91.73%)
Average	0.61%	(16.75%)	1.46%	(17.6%)	8.54%	(20.21%)
Average Lower Bound						
Linear relaxation	9.55%		10.87%		0.00%	
After run	16.77%		16.84%		19.79%	
No solution	7		5		22	
Optimal solution	57		57		52	
Items						
5	0.23%	(6.8%)	0.62%	(7.07%)	1.11%	(9.08%)
10	0.53%	(16.8%)	1.54%	(17.64%)	16.66%	(22.46%)
15	1.11%	(27.36%)	2.27%	(28.69%)	8.26%	(31.32%)
Periods						
20	0.29%	(6.65%)	0.20%	(6.67%)	1.81%	(10.91%)
30	0.78%	(19.99%)	0.99%	(19.87%)	12.90%	(25.84%)
40	0.79%	(24.2%)	3.29%	(26.77%)	11.72%	(25.1%)
Long Setup						
400	0.16%	(4.3%)	0.28%	(4.58%)	0.94%	(6.17%)
700	0.69%	(20.03%)	1.14%	(20.57%)	6.48%	(24.57%)
1200	1.02%	(26.71%)	3.05%	(28.24%)	20.32%	(32.45%)

best model in the literature. The three models provided similar results concerning number of provable optimal solutions found. The superior performance of the proposed models is more significant as the number of items, periods and *MLST* increases.

The differences of *CLSP-BL-SCC* models (Table 2.7) are clearly enhanced. On average, the best performance was achieved by *DSM*, which provided the best solutions, final lower bound and consequently optimality gap in the time limit imposed. *Su08* could not find a solution for 22 out of 270 instances. The literature model showed a weak performance, and the worst case was 6 times worse than the best solution found. For the instances with *MLST* equal to 1200, the *DSM* and *Su08* solutions are 1.02% and 20.32% on average from the best solution found by the three models. In general, both solution and gap performance decrease when the parameters are increased. However, the performance of *DSM* is clearly more robust than that of the literature model.

For both problems (*CLSP-SCC* and *CLSP-BL-SCC*) and the proposed instances, *Su08* provided the best lower bound (using linear relaxation) than the proposed methods. However, after the run, the proposed methods *DSM* and *CMLM* achieved better lower bounds than *Su08*, on average. The average time solutions for all the instances and models are 518.94 and 500.53 seconds for *CLSP-SCC* and *CLSP-BL-SCC*, respectively. The average times of the three models for both problems have a negligible maximum difference of 2.1%.

2.4 Conclusion

Two new formulations have been proposed for the capacitated lot-sizing problem with backlogging and setup carryover and crossover. Both were modelled with the facility location reformulation. *CMLM* combines some elements of Suerie & Stadtler (2003) and Sung & Maravelias (2008). *DSM* offers a new manner of modelling setup variables, which considers both start and end periods of a setup operation. This approach implies a more compact model.

Two problem data sets were generated, considering distinct problem sizes and setup times. The former computational results show that setup crossover is an important modelling feature in case setup times consume a considerable part of the period capacity. The setup crossover also gives the decision-maker flexibility to better utilise the idle capacity, opportunity costs and freedom to choose the period size, independently of the setup times magnitude. The latter computational results show that the proposed formulations outperformed the model from the literature. For *CLSP-SCC*, the best model was *DSM*. It has yielded almost the same average solution of *CMLM*, however it found solutions for more instances and achieved the smallest upper bound on the relative difference against the other models, showing its robustness. For *CLSP-BL-SCC*, the advantage of using *DSM* is enhanced. *DSM* showed the best average solution performance.

Further research can validate and extend this setup modelling technique to other production environments with features, such as parallel lines, sequence-dependent setups and multi-level production systems.

3 CLSP with perishable products

Production planning problems face the challenge of meeting customer demand, respecting the production environment and resource limitations and searching for efficient and cost-saving plans. One of these problems refers to lot sizing which looks for settings of when and how to manufacture products to meet the demand, tackling medium to short term decisions. The capacitated lot-sizing problem (*CLSP*) is defined by a single capacity constrained resource where multi-item production orders should take place, over a planning horizon composed of multiple periods.

Perishable goods represent a challenge to the production planning of many industries. The perishability behaviour of these products implies usefulness and value decrease over time. The definition of perishable products included in Amorim *et al.* (2013b) review is more refined: “A good, which can be a raw material, an intermediate product or a final one, is called “perishable” if during the considered planning period at least one of the following conditions takes place: (1) its physical status worsens noticeably (e.g. by spoilage, decay or depletion), and/or (2) its value decreases in the perception of a(n internal or external) customer, and/or (3) there is a danger of a future reduced functionality in some authority’s opinion”.

The literature presents some real-world examples with perishable goods of industries and services. A first shot on examples would be on food, such as yoghurt (ENTRUP *et al.*, 2005; KOPANOS *et al.*, 2010) and seafood (CAI *et al.*, 2008). However, perishability issues are also found on non-food problems, like blood bank management (MILLARD, 1959), newspaper (BUER *et al.*, 1999) and ready-mix concrete paste (GARCIA; LOZANO, 2004; GARCIA; LOZANO, 2005). In all these problems, the decision maker must handle the trade-off between supply chain costs, ageing and freshness of finished products. For instance, in production planning, the decision maker may decide to aggregate production of multiples demand orders and hold finished products on inventories to save costs, hence resulting in “older” (less fresh) finished products to customers (which may lead to the pricing problem).

The studied problem uses *CLSP* framework along with two main extensions: a) setup carryover is allowed, and; b) products are perishable, with a fixed lifetime, measured in terms of periods. Two novel mathematical formulations are proposed to encompass *CLSP* with setup carryover and perishable products (hereby named as *CLSP-PP*). The former model has lot-sizing and inventory variables traditionally defined, where production and inventory levels are explicitly monitored at the end of every period. In the latter, the lot-sizing variables were redefined using the facility location reformulation (KRARUP; BILDE, 1977). This redefinition provides a simpler way to follow product perishability, since the variable indices track production period along with the demand order met by

that production. A set of instances is generated, based on previous literature instances and solved by a mixed integer linear programming solver (*MILP*-solver) with limited time. Good quality solutions were obtained by the *MILP*-solver for the proposed models, however, there is no guarantee of optimality in most of them, and still a considerable integer gap to shorten.

The remainder of the chapter is organised as follows: Section 3.1 provides a brief literature review; Section 3.2 states the problem and presents two novel models for *CLSP-PP*; Section 3.3 reports the computational tests and; Section 3.4 adds conclusion and some perspectives of research in this topic.

3.1 Literature Review

Perishability is reminded as a hot topic by Clark *et al.* (2011), which presents some extensions and opportunities for lot-sizing research. Perishability is also one extension of the remarkable effort of the operation research society to incorporate more real world specificities of the production environment in their mathematical formulations, as highlighted by Jans & Degraeve (2008). However, perishability may occur in all parts of the supply chain, from procurement of perishable raw materials, through production planning and inventory management until the delivery of perishable finished products to customers.

There are some reviews on literature dedicated to this topic, mainly on inventory management, from Nahmias (1982) to Karaesmen *et al.* (2011). Two recent reviews emphatically address production planning problems with perishability. Amorim *et al.* (2013b) survey focus on production and distribution planning with perishability. A framework to classify perishability was proposed, based in three parameters: a) physical product deterioration (yes/no); b) authority limits (fixed/loose) and; c) customer value (constant/decreasing). Specifically on production planning, problems concerning lot sizing and/or scheduling were addressed. Pahl & Voß (2014) reviewed papers dealing with perishability, its depreciation effects and the modelling of lifetime constraints on supply chain management. Tactical and operational production planning problems such as lot sizing and scheduling were considered, split by deterministic/stochastic models and the planning horizon (finite/infinite approaches). It is highlighted that lot sizing is of paramount importance on determining the lead times, on which we may infer perishable product quality. Regarding the different issues approached by the reviews, our aim is on modelling deterministic lot-sizing problems with fixed lifetime perishable goods, constant customer value and dynamic demand over a finite planning horizon. Closer approaches of the literature on this problem are listed below.

Pahl & Voß (2010) address the lot-sizing and scheduling problem with perishable products of fixed lifetime. The paper introduces one *CLSP* model and two small-bucket formulations, namely the discrete lot-sizing and scheduling problem (*DLSP*) and the

proportional lot-sizing and scheduling problem (*PLSP*). The big-bucket formulation does not consider setup carryover over macro-periods, unlike the latter models. The all-or-nothing assumption of *DLSP* worsens the production plan solutions, causing spoilage and related costs. Therefore, *PLSP* formulation framework seems more suitable to tackle perishability. Pahl *et al.* (2011) extend the previous models assuming sequence-dependent setup times and costs. A *CLSP* model with setup carryover and a general lot-sizing and scheduling problem (*GLSP*) are proposed. Both papers consider spoilage, which may occur because minimum lot size constraints are taken into account. Without that constraint, spoilage will not occur and may be discarded.

Amorim *et al.* (2011) propose two multi-objective formulations to tackle the lot scheduling problem with fixed lifetime perishable items. The proposed formulations rely on well known modelling techniques from the literature: a) *GLSP* for parallel machines; b) simple plant location reformulation for production lot size variables; and c) block planning approach, respectively (MEYR, 2002), (KRARUP; BILDE, 1977) and (GÜNTHER *et al.*, 2006). Two novel formulations were devised with a difference regarding the strategies make-to-order or hybrid make-to-order/make-to-stock. Total costs and freshness compose the multi-objective function. A genetic algorithm was developed, in which the decision maker may explore the set of Pareto non-dominated solutions to choose the best balance. According to Amorim *et al.* (2011), freshness measures customer satisfaction, mainly for perishable products with physical deterioration as foods. Amorim *et al.* (2012) extend the above work proposing novel models for the integrated production and distribution planning problem of perishable products, maintaining the multi-objective framework. Furthermore, coupled/decoupled approaches and fixed/loose shelf-life of perishable items are tested. Amorim *et al.* (2012) highlight that the joint production and distribution planning decisions dominates the decoupled approach for an illustrative example.

Caserta & Voß (2013) use the concept of perishability to improve a solution method for *CLSP*. The authors look to perishability constraints as hop constraints, in the sense that each demand order must be met with a limited number of arcs, considering the production flow scheme (production and incoming inventory is equal to demand and ongoing inventory). The method combines Dantzig-Wolfe decomposition with a metaheuristic to obtain good columns, for *CLSP* without perishability constraints. Perishability is inserted in the approach by fixing products shelf-lives to zero periods and iteratively increasing the shelf-life until a criteria is met. The search space is first limited with product perishability and further relaxed gradually, returning to the original problem. Although the method was designed for *CLSP* without perishable products, the proposed method clearly fits the perishable case. However, few tests were made (just six instances of Trigeiro *et al.* (1989)) and the results lack of a sensitivity analysis, such as considering more values for *TBO* (time between orders). Moreover, it is not clear how the proposed method will face problems with heterogeneous demand behaviour, for instance, in the case of seasonality,

unlike the uniform demand proposed by Trigeiro *et al.* (1989).

3.2 Problem statement and proposed models

In this section, we state the problem and propose two formulations to the capacitated lot-sizing problem with setup carryover and perishable products.

The *CLSP-PP* consists of planning and scheduling production lots of perishable products to meet a known demand in a planning horizon. The planning horizon is composed of T uniform periods with a fixed capacity time. To manufacture a production order of an item, the single production unit (machine or line) has to be appropriately set up. The setup operations are sequence-independent, incur on some costs and spend capacity time. The line remains set up for an item until a new setup operation is made, even when changing periods, *i.e.*, the setup state is carried over adjacent periods. The setup carryover feature is present in many production environments, such as 24/7 lines, and directly affects the production planning, since it may reduce one setup operation per period, consequently saving time and costs. Perishable products are known for having a decreasing utility over time. In this case, the perishable items have a fixed lifetime accounted after the products are finished, measured in periods. For instance, a shelf-life of two periods means that the finished product remains useful and marketable for the current period and the following two. In other words, a demand order for an item in a period may be produced at most two periods ahead.

The proposed models are based on Suerie & Stadtler (2003) and Sahling *et al.* (2009) formulations. The first model is based on the classical definition of the production size variables (X_{it}), in which all production of an item in the current period is aggregated, no matter which the demand orders satisfied by that production. At the end of a period, the demand is met using inventoried items from previous periods and the production from the current one. The remaining finished products are hold to the following periods. This aggregated approach requires a FIFO policy (first-in-first-out) at the inventory to avoid spoiled products. In the second model, the facility location variable reformulation proposed by Krarup & Bilde (1977) is adopted. The variable reformulation provides a clever use of the production size variable ($X_{itt'}$), where the variable tracks the production (t) and the demand order (t') dates for item i . Therefore, some variables $X_{itt'}$ are eliminated, because the production order to meet demand of item i in period t' is restricted to periods $t', t' - 1, \dots, t' - sl_i$, where sl_i stands for the shelf-life of product i (note that backlogging is not allowed). The second model provides a disaggregated way to see the production lot sizes, instead of the variables used in the first formulation. The relation between the production size variables used by the approaches is given by $X_{it} = \sum_{t'=t}^T d_{it'} X_{itt'}$. The first and second models will be appointed as classic (*CF*) and facility location (*FLF*) formulations, respectively. The indices, parameters and variables necessary to the mathematical

models are defined in the following:

Indices

i	products (items)
t, t'	periods

Parameters

N	number of items, also represent the set of items
T	number of periods, also represent the set of periods
hc_i	holding cost of item i per unit per period
sc_i	setup cost for item i
pt_i	processing time of item i per unit
st_i	setup time for item i
sl_i	shelf-life of product i (in multiples of periods)
cap_t	capacity of line in period t (in time units)
d_{it}	demand for item i in period t

Decision Variables

X_{it}	production lot size for item i in period t
I_{it}	inventory level for item i at the end of period t
$X_{itt'}$	fraction of the demand satisfied for item i in period t' produced in period t
S_{it}	equals 1 if setup state i is active in period t (0 otherwise)
α_{it}	equals 1 if setup state i is active at the beginning of period t (0 otherwise)
Q_{it}	equals 1 if only setup state i is present in period t (0 otherwise)

The first proposed mathematical model (aggregated formulation – CF) reads:

$$\text{Min } \sum_{i=1}^N \sum_{t=1}^T hc_i I_{it} + \sum_{i=1}^N \sum_{t=1}^T sc_i (S_{it} - \alpha_{it}), \quad (3.1)$$

$$\text{s.t. } I_{i,t-1} + X_{it} = d_{it} + I_{it}, \quad \forall i, t, \quad (3.2)$$

$$\sum_{t'=1}^{t-sl_i} X_{it'} \leq \sum_{t'=1}^t d_{it'}, \quad \forall i, t, \mid t > sl_i, \quad (3.3)$$

$$\sum_{i=1}^N pt_i X_{it} + \sum_{i=1}^N st_i (S_{it} - \alpha_{it}) \leq cap_t, \quad \forall t, \quad (3.4)$$

$$X_{it} \leq \min \left(\frac{cap_t}{pt_i}, \sum_{t'=t}^{\min\{T, t+sl_i\}} d_{it'} \right) S_{it}, \quad \forall i, t, \quad (3.5)$$

$$\sum_{i=1}^N \alpha_{it} \leq 1, \quad \forall t, \quad (3.6)$$

$$\alpha_{it} \leq S_{i,t-1}, \quad \forall i, t, \quad (3.7)$$

$$\alpha_{it} \leq S_{it}, \quad \forall i, t, \quad (3.8)$$

$$\alpha_{i,t+1} + \alpha_{i,t} \leq S_{it} + Q_{it}, \quad \forall i, t, \quad (3.9)$$

$$(S_{it} - \alpha_{it}) + \sum_{j=1}^N Q_{jt} \leq 1, \quad \forall i, t, \quad (3.10)$$

$$Q_{it} \leq \alpha_{it}, \quad \forall i, t, \quad (3.11)$$

$$Q_{it} \leq \alpha_{i,t+1}, \quad \forall i, t, \quad (3.12)$$

$$S_{it}, \alpha_{it} \in \{0, 1\}, \quad \forall i, t. \quad (3.13)$$

$$0 \leq X_{it}, Q_{it} \leq 1, \quad \forall i, t. \quad (3.14)$$

The objective function (3.1) minimises the sum of holding costs and setup costs. Equations (3.2) determine the flow balance between production level, incoming/outgoing inventory and demand satisfaction for each period. Constraints (3.3) impose that spoilage is forbidden, since the total production level from the first period to the current period should be less or equal to the sum of the doable demand orders. Capacity constraints (3.4) limit production and setup operations in a period. Production of an item in a period may occur only if the line is set up (3.5). Moreover, it is upper bounded by the capacity of the line and the doable demand orders. Constraints (3.6) imply that at most one setup state is carried over between consecutive periods. In case the setup state is preserved between periods $t - 1$ and t , constraints (3.7) and (3.8) force the existence of setup state i in the respective periods. Inequalities (3.9) and (3.10) determine consecutive setup carryovers. The former constraints require that setup state should be carried over from period $t - 1$ to period t and then to period $t + 1$ and the latter implies that the setup of neither item occurs in period t . Constraints (3.11) and (3.12) impose that variables Q_{it} are positive only if α_{it} and $\alpha_{i,t+1}$ are positive. Consequently, by considering the sum of constraints (3.11) over items, we have $\sum_{i=1}^N Q_{it} \leq \sum_{i=1}^N \alpha_{it} = 1$. Therefore, only one setup state may be maintained over two consecutive periods. The remaining constraints state the variable

domain. Although variables Q_{it} are defined as a binary variable, they do not need to be defined explicitly in the model.

The disaggregated formulation (*FLF*) is defined below:

$$\text{Min} \sum_{i=1}^N \sum_{t=1}^T \sum_{t'=t}^{\min\{T, t+sl_i\}} hc_i(t' - t) d_{it'} X_{itt'} + \sum_{i=1}^N \sum_{t=1}^T sc_i(S_{it} - \alpha_{it}), \quad (3.15)$$

$$\text{s.t.} \quad \sum_{t'=\max\{1, t-sl_i\}}^t X_{itt'} = 1, \quad \forall i, t \mid d_{it} > 0, \quad (3.16)$$

$$\sum_{i=1}^N \sum_{t'=t}^{\min\{T, t+sl_i\}} pt_i d_{it'} X_{itt'} + \sum_{i=1}^N st_i(S_{it} - \alpha_{it}) \leq cap_t, \quad \forall t, \quad (3.17)$$

$$X_{itt'} \leq S_{it}, \quad \forall i, t, t' \in t, \dots, \min\{T, t + sl_i\}, \quad (3.18)$$

$$0 \leq X_{itt'} \leq 1, \quad \forall i, t, t', \quad (3.19)$$

$$(3.6) - (3.14).$$

The objective function (3.15) is analogous to (3.1) and minimizes holding and setup costs. Equations (3.16) ensure that the entire planning horizon demand is met without any backlog. Notice that these constraints and the production variable redefinition replace constraints (3.2) and (3.3). Capacity constraints are given by (3.17) and the imposition of line setup state for any production is provided by (3.18). The new production variable domain is given by (3.19). The remaining constraints (3.6) - (3.14) hold the definition and relations between setup carryover, consecutive setup carryover and setup states.

3.2.1 Example

The numerical example of Trigeiro *et al.* (1989) is rewritten, with some modifications. The problem has 3 items and 4 periods. Capacity per period is equal to 175 time units and production times are equal to 1 time unit per item produced. The remaining data is given in Table 3.1. The changes in the literature example remains in the perishability of the items and the different holding cost values. Notice that, if the shelf-life of an item has the same size of the number of periods T , it means that for that planning horizon, the production of that item is not constrained/affected by perishability issues.

The optimal solution of this example is illustrated in Figure 3.1, which is a Gantt chart representation of the production and setup operations through the planning horizon. Each bar denotes a period of the planning horizon. The setup operations are in black colour and the production operations are in white colour with the respective production variables $X_{itt'}$ from *FLF* and the aggregated production value. It is worth noticing that the production variables of *CF* (X_{it}) are obtained by aggregating production variables of

Table 3.1 – Remaining data of the example.

Item	Setup Time	Setup Cost	Holding Cost	Shelf-life	Demand by period			
					1	2	3	4
1	15	60	2	1	40	45	65	35
2	10	120	1	1	35	35	55	35
3	15	80	2	1	0	60	45	80

the FLF ($\sum_{t'=t}^T d_{it'} X_{itt'}$). From the chart is possible to recognize that the setup state is carried over from the current period to the respective subsequent.

To verify the trade-off imposed by the perishability over cost functions, the products are then considered not perishable. Hence, a different optimal solution is achieved (Figure 3.2). Notice that in the “relaxed” solution the entire production of item 2 in periods 3 and 4 are anticipated to period 2. Moreover, the number of setup operations is reduced from 7 to 5. Short shelf-lives may push more setup operations for the production plans as the instance exhibit. In a critical case, a shelf-life equal to zero would impose lot-for-lot solutions, *i.e.*, products should be manufactured in the same period demand is met.

Figure 3.1 – Optimal solution to the $CLSP-PP$ example (660 cost units).

Figure 3.2 – Optimal solution to the $CLSP-PP$ example relaxing shelf-life constraints (640 cost units).

3.2.2 Valid Inequalities

Some valid inequalities are pointed out by Suerie & Stadtler (2003) and they are adapted to the perishable products case. The inequalities are derived from the problem data and are focused on establishing that some Q_{it} variables must be equal to zero. The main idea is that the cumulative demand production times and some must-have setup times imply that a period should produce at least two products, and consequently, variables Q_{it} become null for all the items of that period. Perishability tightens these constraints because multiples setups must be considered. Without loss of generality, we may consider that there is positive demand for all items and periods. Be item i with shelf-life sl_i and a planning horizon of T periods. Disregarding setup carryover, the minimum number of setups for item i in the planning horizon is given by $\left\lceil \frac{T}{sl_i+1} \right\rceil$. So, in case $sl_i = 0$, T setups are needed and if $sl_i \geq T - 1$, just one setup might be sufficient and all demand orders of item i may be produced in the first period. Now, be $mincumpt_t$ the minimum cumulative production time from period 1 to t , given by Equation (3.20). It considers the production time to meet all the demand until period t and the necessary setup times for each product. The number of setup operations that should be made from period 1 to t due to shelf-life constraints is given by $\left\lceil \frac{t}{sl_i+1} \right\rceil$. For instance, a product with shelf-life equal to one period should have at least one setup every two periods, *i.e.*, as inventory is hold for at most one period, new setup operations are needed to manufacture the items. However, as there is setup carryover, some setups may be unnecessary and so, $mincumpt_t$ is decreased by the maximum setup time. These setup reductions are accounted only when second setups for products are required, *i.e.*, from period $\min_{i \in N} sl_i + 1$ onwards.

$$mincumpt_t = \sum_{i=1}^N \sum_{t'=1}^t d_{it'} p_{t'} + \sum_{i=1}^N \left(st_i * \left\lceil \frac{t}{sl_i+1} \right\rceil \right) - \max \left(0, \left(t - 1 - \min_{i \in N} sl_i \right) \max_{i \in N} st_i \right) \quad (3.20)$$

An instance is feasible when $\sum_{t'=1}^t cap_{t'} \geq mincumpt_t$, $\forall t$, *i.e.*, the cumulated capacity is sufficient to fit the entire production and some setups, in the most compact configuration. If $\sum_{t'=1}^{t-1} cap_{t'} \geq mincumpt_t$, then period t might be idle. Therefore, in case the cumulated capacity until period $t - 1$ is sufficient to fit the minimum cumulative production time from period 1 to t , without considering the production of demand order d_{it} , then period t might have $Q_{it} = 1$. Otherwise, $Q_{it} = 0$. The inequality (3.21) defines how the variable Q_{it} is upper bounded:

$$Q_{it} \leq \frac{\sum_{t'=1}^t cap_{t'} - mincumpt_t + p_{t'} d_{it}}{cap_t}, \quad \forall t > 2. \quad (3.21)$$

In the first period, $Q_{i1} = 0, \forall i$ and the capacity of period 1 should be greater than or equal to $mincumpt_1$ ($cap_1 \geq ct_1$). Now, consider the planning horizon until the second period. The aim is to find that neither setup state may be maintained from period 1 to 2 nor from period 2 to 3 ($\alpha_{i2} + \alpha_{i3} \leq 1$, $\forall i \in N$). Therefore, it suffices to confirm

that $\sum_{i'=1}^1 cap_{i'} - mincumpt_2 - \max_i d_{i2}p_i$, $\forall i \in N$ is negative. If so, $Q_{i2} = 0$, $\forall i \in N$. Inequalities (3.21) are successful for problems with tight resource capacities, as the minimum cumulative production time tends to be bigger than the sum of the capacity of the previous periods. In case Q_{it} is forced to have a negative value, the problem is naturally infeasible.

Considering the example of Subsection 3.2.1, we obtain $mincumpt_t$ equal to 255 and 445 time units for $t = 2$ and $t = 3$, respectively and $Q_{i1} = 0$ and $Q_{iT} = Q_{i4} = 0$, by definition. So, using the rule (3.21):

$$\begin{aligned} Q_{i2} &\leq \frac{cap_1 + cap_2 - mincumpt_2 + pt_i d_{i2}}{cap_2} = \frac{95 + d_{i2}}{175}, \quad \forall i, \\ Q_{i3} &\leq \frac{cap_1 + cap_2 + cap_3 - mincumpt_3 + pt_i d_{i3}}{cap_2} = \frac{80 + d_{i3}}{175}, \quad \forall i, \end{aligned}$$

and then, all variables Q are equal to zero, *i.e.*, neither setup state will be maintained over a period, since it is indispensable at least two setup states for each period.

3.3 Computational experiments

Computational tests are run for both proposed formulations. The data set is based on literature sets of instances, with some adaptations to the perishable setting. The computational results are compared and shown in the following.

3.3.1 Data

The data set is based on the papers of Trigeiro *et al.* (1989), Süral *et al.* (2009) and Muller *et al.* (2012). Süral *et al.* (2009) chose some of the most difficult instances generated by Trigeiro *et al.* (1989), belonging to data subset G: instances G51 to G60 and G66 to G75, being five instances for each combination size ($N \times T$) of 12×15 , 24×15 , 12×30 and 24×30 . Muller *et al.* (2012) expanded the data set by concatenating instances, generating instances with larger planning horizons. The procedure is simple and the new instances are generated by taking the original and aggregating the same instance to the planning horizon two and three times. Therefore, from an instance whose combination is 24×30 they created instances whose combination 24×60 and 24×90 ($d_{it} = d_{i,t+30} = d_{i,t+60}$, $\forall i = 1, \dots, 24$, $t = 1, \dots, 30$). For our study, we assume the same instances which have inspired Süral *et al.* (2009) and Muller *et al.* (2012) data sets. The data set is expanded using the idea of concatenating instances of Muller *et al.* (2012). However, we have increased the planning horizon and the number of products of the original instances. For example, the resulting problems for a 24×30 instance is 48×30 , 72×30 instances and 144×30 instances ($d_{it} = d_{i+24,t} = d_{i+48,t} = d_{i+72,t} = d_{i+96,t} = d_{i+120,t}$, $\forall i = 1, \dots, 24$, $t = 1, \dots, 30$). In this case, the resource capacity of period t of the new instances should also be increased by doubling or tripling cap_t . The resulting instances have all the combinations of 12, 24,

48, 72 and 144 products and 15, 30, 60 and 90 period planning horizons, summing up 20 different instance sizes.

Until now, perishability issues were not taken into account for the above instances. Furthermore, perishable products are assumed and different shelf-life durations considered, according to the classification: short, medium, variable and original. In short and medium shelf-life instances, all perishable products have a lifespan duration of 2 to 3 periods and 4 to 8 periods, respectively. Original perishable products have a shelf-life duration greater than the planning horizon, like the original instances, where no perishability is assumed. The variable shelf-life instances have 25%, 25% and 50% of the products classified as short, medium and original, respectively. Lastly, we propose a holding cost reduction of 75% to measure the effect of these costs, since the willingness to stock increases, although the product freshness is decreased on demand due date. Thus, five instances per class of five different number of products, four number of periods, four perishable product shelf-life durations and two holding cost structures, results in 800 distinct instances.

3.3.2 Computational results

All computational experiments were performed on an Intel Core i5 processor, with 2.80 GHz CPU and 8GB RAM under Linux Ubuntu 10.04 (64 bit). CPLEX version 12.6 from IBM was used as the MIP solver. The data generator described above was used to obtain the instance set. Three tests were performed using different computational times to solve each *MILP* formulation and the parallel mode was active (4 threads). These computational times were limited to 60, 600 and 1800 seconds. Therefore, six approaches are compared, denoted by *CF60*, *CF600*, *CF1800*, *FLF60*, *FLF600* and *FLF1800*, representing the method and time running limit, respectively.

Table 3.2 shows the average optimality gap of the set of instances, comparing the three rounds of tests (columns 60s, 600s and 1800s) for *CF* and *FLF*. Results are aggregated by instance types, based on instance parameters as number of items, periods and the perishability and holding cost structures. The optimality gap is the relative difference between the incumbent solution (*incsol*) value and the best lower bound (*incLB*) achieved ($optgap = \frac{incsol - incLB}{incsol}$). The first rows of the table indicate the average optimality gap and the worst optimality gap obtained so far. Then, the number of instances for which an integer solution was not achieved and those which were proven optimal are shown. In the following, the average computational times in seconds are compared. The next rows detail the optimality gaps according to: the number of items (12, 24, 48, 72 and 144); the number of periods (15, 30, 60 and 90); the perishability structure (short, medium, variable and original shelf-life) and; holding cost structure (original 100% costs and reduced 25% costs).

Table 3.2 explicits some obvious results, as the more time is given to the MILP-solver, the better the optimality gap results are, on average. The MILP-solver was not able to

find feasible solutions for one instance out of 800 for neither method. The other instances had at least one solution achieved by at least one formulation. Most of the instances were not solved to optimality (gaps less than 0.05%) considering the time limits by any method (711 out of 800). Hence, the average computational times are close to the time limit imposed for the tests, with *FLF* approaches slightly better. Regarding the number of items, the optimality gap trend of the two models is not the same. As the number of items increases, the optimality gap increases for *CF* while decreases for *FLF*. This might be explained by the tightness of the latter formulation, in which lot-sizing decision and setup state relation depend only on the demand order size (3.18), unlike the first model, which depends on line capacity (3.5), clearly enlarged for greater number of items. It is well known in the lot-sizing literature that problems with more periods tend to present worse optimality gaps, which is confirmed in this study. Shorter lifetimes of the perishable products tight the problem, improving the optimality gap performance mainly for *FLF*. It is noteworthy that variable perishability structure has products with short and medium shelf-life, and the performance seems close to medium perishability structure. Moreover, for *FLF*, longer shelf-lives increase the number of production lot variables, since the production of a demand order might be processed in earlier periods, which shows that *FLF* is more sensitive than *CF* to this instance parameter. As expected, the final optimality gap of the instances increases for smaller holding costs. Reduced holding costs means that pushing finished products to the inventory is more attractive, the opposite of what perishability suggests. So, in terms of optimality gap, the instances are harder to solve for reduced holding costs.

Figure 3.3 exhibits a performance chart for optimality gap, based on Dolan & Moré (2002) (Appendix A). In this case, the chart shows the cumulative frequency curve of the optimality gaps of the two formulations and the three runs of each formulation. *CF* and *FLF* are denoted by the double grey and single black curves, respectively, and the different running time limits are denoted by the line dash style. The better curve is the one that achieves higher frequency of optimality gaps within smaller range, starting from zero, *i.e.*, the curve which looks closer to the top left corner of the chart. Therefore, the best approach is *FLF1800*, which obtained approximately 88% of the instances tested with an optimality gap less or equal 10%. It is important to notice that *CF1800* found solutions to more instances than *FLF1800*, however with higher optimality gaps. In the following, the solution values are also compared.

Table 3.3 reports the average of the relative difference of the solution found in each run, over the best solution achieved in all the runs. Be $incsol_{\alpha,\beta}$ the incumbent solution for approach α to instance β and $bestsol_{\beta}$ the best solution achieved by all the approaches. Therefore, the relative difference $solgap_{\alpha,\beta}$ for approach α and instance β is calculated by

$$solgap_{\alpha,\beta} = \frac{incsol_{\alpha,\beta} - bestsol_{\beta}}{bestsol_{\beta}}.$$

Table 3.2 – Optimality gaps for *CF* and *FLF*.

	<i>CF</i>			<i>FLF</i>		
	60s	600s	1800s	60s	600s	1800s
Average	16.07%	9.62%	6.55%	8.03%	4.81%	3.79%
Maximum	77.80%	75.95%	63.05%	68.42%	65.46%	65.45%
No solution	138	17	2	122	31	11
Optimal	28	67	73	46	74	85
Time (seconds)	59	573	1696	58	565	1678
Items						
12	12.16%	7.52%	6.86%	11.95%	7.14%	6.30%
24	14.71%	7.58%	4.78%	10.07%	4.91%	4.20%
48	16.85%	10.00%	6.05%	6.17%	4.55%	3.17%
72	19.22%	10.24%	6.56%	5.98%	4.07%	3.18%
144	20.60%	13.03%	8.51%	3.64%	3.11%	1.97%
Periods						
15	2.40%	1.01%	0.88%	1.62%	0.92%	0.78%
30	13.20%	4.84%	3.20%	5.52%	3.19%	2.82%
60	25.65%	14.12%	8.47%	13.72%	6.13%	4.57%
90	30.04%	19.30%	13.72%	15.11%	9.60%	7.15%
Perishability						
S	14.43%	8.31%	5.35%	5.26%	3.06%	2.74%
V	17.06%	9.75%	7.20%	7.86%	3.82%	3.28%
M	16.27%	10.26%	7.12%	8.60%	4.51%	3.67%
O	16.73%	10.19%	6.52%	10.83%	7.91%	5.46%
Holding Cost						
100%	6.39%	2.05%	1.33%	2.47%	1.13%	0.87%
25%	26.42%	17.36%	11.79%	14.86%	8.74%	6.79%

Figure 3.3 – Performance chart for optimality gap.

Hereafter, the measure *solgap* is called as solution gap. Table 3.3 structure and display is analogous to Table 3.2. Again, the results indicate that, on average, solutions are improved as more time is given to the MILP-solver. However, due to the MILP-solver stochastic behaviour, sometimes with less time a better solution may be achieved. In the larger case, the solution of *CF1800* is 38.9% greater than solution of *CF600* and, for *FLF*, the larger case has *FLF1800* is 0.3% greater than solution of *FLF600*. Although the maximum solution gap on *FLF600* is greater than on *FLF60*, it is worth noticing that more instances have a solution found, so it is reasonable to have more distant solutions to the best solution, even with more time. The average solution gap of *FLF* is better than *CF* for all running time limits and the comparison over the parameters. The *FLF* also seems more robust than the other approaches, since the average solution gap remains below 2.5%.

Table 3.3 – Average relative difference over solutions for *CLSP-PP*.

	<i>CF</i>			<i>FLF</i>		
	60s	600s	1800s	60s	600s	1800s
Average	21.07%	10.13%	4.21%	8.49%	2.35%	0.66%
Maximum	157.99%	153.32%	153.32%	145.86%	166.74%	166.74%
Items						
12	6.90%	0.82%	0.25%	9.39%	1.21%	0.16%
24	17.35%	5.73%	0.59%	10.87%	1.07%	0.07%
48	26.29%	13.47%	5.91%	5.80%	3.96%	1.11%
72	29.06%	14.32%	5.79%	8.86%	3.45%	1.65%
144	35.91%	16.97%	8.54%	6.53%	2.12%	0.30%
Periods						
15	1.42%	0.06%	0.03%	0.33%	0.03%	0.01%
30	16.64%	2.31%	0.13%	2.92%	0.34%	0.05%
60	37.18%	16.67%	6.29%	17.55%	3.33%	0.62%
90	38.49%	22.50%	10.44%	18.39%	6.18%	2.03%
Perishability						
S	17.31%	7.86%	3.27%	2.94%	0.28%	0.03%
V	24.60%	10.67%	5.08%	8.35%	0.62%	0.07%
M	22.16%	11.23%	5.39%	8.76%	1.23%	0.24%
O	20.63%	10.83%	3.07%	14.81%	7.38%	2.33%
Holding Cost						
100%	6.90%	1.33%	0.50%	1.73%	0.30%	0.05%
25%	36.21%	19.13%	7.93%	16.65%	4.54%	1.29%

Figure 3.4 is again a Dolan-Moré chart, which illustrates the outperforming of *FLF1800* over the other approaches. The chart shows the cumulative frequency curve of the solution gaps. The chart was trimmed for solution gaps greater than 100%. Notice that more than half of the instances have solution gaps lower than 5%. In that range, even *FLF600* have more solutions closer to the best solution found than *CF1800* does. Figures 3.3 and 3.4 corroborates on showing the victory of *FLF1800* approach over the others. However, two weaknesses should be highlighted: a) the number of instances without a solution found

(11 out 800 and worse than *CF1800*) and; b) the optimality gap (3.79% on average and 65.45% on the worst case), which may be shorten.

Figure 3.4 – Performance chart for solution gap.

3.4 Conclusion

The capacitated lot-sizing problem with setup carryover and perishable items was defined and explored in this chapter. Medium and short term decisions are tackled together to obtain a production plan regarding perishability issues. Few literature studies were found on this topic and managing perishable goods in production environments is challenging.

Two models were proposed, assuming two different production lot size variable definitions: the classic and the facility location reformulation. The latter, in a certain way, tags the production of an item to its demand order, defining simultaneously the production amount and the quality of demand met. A set of instances was created for computational tests, according to other literature instance data generators. Computational results reported that even for half an hour MILP-solver runs, most of the instances were not solved to optimality and there was even one case where no solution was achieved by any proposed approach. For shorten times as 60 seconds, approximately 17% and 15% of the instances have no solution as answer for *CF60* and *FLF60*, respectively.

Although the results indicate a medium to good performance of the MILP-solver for greater computational times, faster results of good quality might be requested, with more reliable approaches (in terms of finding feasible solutions). Thus, it is reasonable to propose other approaches to the problem, as the use of heuristics and metaheuristics to improve reliability and speed in achieving solutions. The model only focuses on the production planning problem, however it is important to develop integrated approaches

to take decisions regarding other aspects of the supply chain, as logistics for distribution of the final products and raw material requirements. It is worth mentioning that even raw materials might be perishable, with a clear impact over the subsequent decisions, including manufacturing and delivering decisions.

4 Lagrangean heuristic for CLSP-PP

As seen in Chapter 3, the capacitated lot-sizing problem with setup carryover and perishable products (*CLSP-PP*) is a challenging problem. There, two novel mathematical formulations were proposed and computational tests were performed for a set of instances based on the literature. Although the MILP-solver achieved good solutions for many instances, neither formulation delivered solutions for all instances. Moreover, solution optimality was not proven for most of the cases and, for some instances, the optimality gap reached more than 50%, even for half-hour runs. Notice that the MILP-solver used is a state-of-the-art optimization software, developed for a large variety of mathematical programming problems. In this sense, there is a lack for a problem-driven solution, with more robustness in terms of achieving proven good-quality solutions for all instances, in less amount of time.

We propose a heuristic approach based on lagrangean relaxation, subgradient optimization and feasibility heuristics to tackle the problem. The main formulation may have capacity constraints and other item-coupling constraints relaxed and, as consequence, independent subproblems concerning each item are obtained. Dynamic programming procedures achieve the optimal solution to these subproblems. The lagrangean relaxation provides the lower bound for the method, complemented by heuristic procedures to find feasible solutions and promote local search. Although the method is not exact, the quality of the solution may be measured, which is a quite important feature of this type of approach.

The remaining of this chapter is given in the following. Section 4.1 addresses the literature on lot-sizing problems with lagrangean-based solution approaches. Section 4.2 restates *CLSP-PP* and the best formulation of Chapter 3. Section 4.3 discusses the details of the lagrangean relaxation (Subsection 4.3.1), subgradient optimization (Subsection 4.3.2) and the feasibility heuristic (Subsection 4.3.3), respectively. The computational results and the comparison against the MILP-solver solutions are provided in Section 4.4. This chapter is concluded in Section 4.5, along with some future research directions.

4.1 Literature review

Lagrangean relaxation is an optimization technique in which “hard constraints” are relaxed from the original problem, changing the feasibility region and providing an “easier” new problem. The omitted constraints are introduced into the objective function with some penalties (lagrangean multipliers) to punish solution infeasibilities. For each set of multipliers, the optimal solutions to the relaxed problem are bounds to the original problem. Therefore, to obtain the best bound, it is necessary to achieve the best lagrangean

multipliers (lagrangean dual problem). To solve the lagrangean dual problem, we use subgradient optimization (*SG*), an iterative procedure which obtains better solutions using the current ones and subgradient directions. The lagrangean relaxation combined with subgradient optimization is largely used in optimization literature. Although the lagrangean relaxation provides only a bound to the problem, the solution of the relaxed problem is a good start for achieving feasible solutions of the original problem. A feasibility procedure should be applied in order to find feasible solutions. The reader interested in the theory and application of the lagrangean relaxation is referred to Geoffrion (1974), Held *et al.* (1974), Shapiro (1979), Fisher (1985), Lemaréchal (2001), Guignard (2003), Fisher (2004) and Lemaréchal (2007).

Lagrangean relaxation approaches have been successfully applied to lot-sizing problems. Thizy & Van Wassenhove (1985) proposed a lagrangean relaxation of the capacity constraints for *CLSP* without setup times. The update of the lagrangean multipliers is provided by subgradient optimization (HELD *et al.*, 1974). The subproblem is solved by Wagner & Whitin (1958) dynamic programming algorithm (*WW*) and the feasibility procedure corresponds to fixing setup variables as obtained by *WW* and solving the reduced problem as a transportation problem (BOWMAN, 1956).

Trigeiro *et al.* (1989) applied the lagrangean relaxation of the capacity constraints to the *CLSP* with setup times. The relaxed problem is solved by *WW*. Then, the feasibility procedure (*TTM*) reduces overtime by shifting and splitting scheduled lots of the solution to the relaxed problem. The *TTM* heuristic provides good feasible solutions, and is composed of five main passes: (1) first backward pass, that attempts to reduce overtime by shifting the production of certain items to earlier periods; (2) first forward pass, whose aim is on eliminating cumulative overtime and shifting inventoried items to later periods (backlogging is forbidden); (3) second backward pass, equal to the first pass; (4) second forward pass, analogous to the first forward pass, except that it tries to eliminate overtime at each period; and (5) fix-up pass, which is applied to the solution to improve the production lot sizes.

Lozano *et al.* (1991) addressed the lagrangean relaxation of the *CLSP* using a primal-dual heuristic based on the formulation proposed by Manne (1958) (set covering approach). The capacity constraints are relaxed and the resulting problem is solvable by *WW*. A restricted linear problem is solved via simplex algorithm to find feasible solution plans. If a solution plan is not found, an ascent direction is devised, based on the restricted dual problem. Two variants are proposed, due to two different approaches to obtain the step size on the ascent direction. However, it is not always possible to find a feasible solution by this method, being necessary other feasibility procedure. Therefore, a lot-shifting heuristic was developed, with three routines, which shifts production for: (1) later periods with available capacity; (2) previous periods in which the setup already exists and slack capacity; and (3) previous periods with idle capacity and no setup.

Diaby *et al.* (1992a) proposed branch-and-bound procedures in which the bounds are provided by lagrangean relaxation and subgradient optimization for the *CLSP* with overtime. The lagrangean relaxations regard capacity or demand constraints. For the former, the subproblem is solved by *WW*. For the latter, the resulting problem is split by periods, where the linear programming relaxation can be seen as a bounded continuous knapsack problem, solved efficiently by a linear programming based branch-and-bound procedure. The leaf nodes of the branch-and-bound tree are solved as transportation problems. Different branching strategies were compared with a better performance of the procedures in which capacity constraints were relaxed. In Diaby *et al.* (1992b) a *CLSP* with multiple resources (regular time and overtime, for example) is solved by a lagrangean relaxation heuristic. The lagrangean relaxation of capacity constraints and subgradient optimization are considered. The feasibility procedure is similar to the transportation problem proposed by Thizy & Van Wassenhove (1985) except that a perturbation scheme is performed to obtain new feasible production plans. The approach is tested on literature problems and on very large scale problems, with up to 5000 products and 30 periods.

Millar & Yang (1993) and Millar & Yang (1994) developed lagrangean decomposition approaches for the *CLSP* without setup times, though the latter considers backlogging. In the decomposition scheme, a copy lot-size variable is introduced and a lagrangean relaxation of the copy constraints is performed, to decompose the problem in two subproblems. A subproblem may be decomposed in N (number of items) uncapacitated lot-sizing problems, solved by *WW* and the other subproblem is a transportation problem, with manageable costs. Comparing the Thizy & Van Wassenhove (1985) feasibility procedure, in which setups found by *WW* are fixed, in this proposed approach, the costs of the transportation problem are modified to discourage production plans in which setup was not found by *WW*, forbidding infeasible solutions. In Millar & Yang (1994), a lagrangean relaxation of the capacity constraints heuristic is proposed, analogous the Thizy & Van Wassenhove (1985), though the resulting transportation problem also have manageable costs. The algorithms apply subgradient optimization to update the lagrangean multipliers.

Tempelmeier & Derstroff (1996) proposed a lagrangean relaxation heuristic approach for the multi-level *CLSP*. The capacity and the multi-level inventory balance constraints are relaxed, resulting in subproblems solvable by dynamic programming algorithm, which provides lower bounds to the original problem. The lagrangean multipliers are updated using subgradient optimization algorithm. The feasibility procedure first ensures the inventory balance constraints, starting from end items to predecessor levels and then overloaded periods are eliminated by shifting their production to periods with slack capacity.

Sox & Gao (1999) presented some models for the *CLSP* without setup times and with setup carryover, regarding classical models and network reformulation models (EPPEN; MARTIN, 1987). A lagrangean relaxation of the capacity constraints and single setup

carryover constraints is proposed to obtain near-optimal solutions. Moreover, the problem is simplified in order to neglect solutions with consecutive setup carryover. The resulting problem is decomposed in N uncapacitated subproblems, where a dynamic programming procedure is proposed. Subgradient optimization is used to define search directions for the next iteration of lagrangean multipliers. The feasibility procedure proposed shifts production to get rid of overloaded periods and limit to a single setup carried over in each period. Briskorn (2006) published a note claiming that the dynamic programming procedure of Sox & Gao (1999) ignores some feasible solutions and proposed a corrected one.

Özdamar & Barbarosoglu (2000) addressed the multi-level *CLSP* with two lagrangean relaxation approaches and subgradient optimization. The first relaxation relaxes only capacity constraints (hierarchical relaxation) and the latter relaxes both inventory balance and capacity constraints (non-restricted relaxation). The resulting subproblems are not optimally solved, so no lower bounds are provided. The feasibility procedure utilises shift production moves in a simulated annealing framework (*SA*), which is also used to perform local search for feasible solutions. The procedures were compared to those of Tempelmeier & Derstroff (1996), with hierarchical relaxation yielding better performance.

The procedure of Hindi *et al.* (2003) is analogous to Trigeiro *et al.* (1989) approach, using the same lower bound approach. However, a more elaborated upper bound procedure is made, using the first four passes of *TTM*. Between these passes, the current setup decision variables are fixed and a transshipment subproblem is solved in attempt to obtain feasible plans. If a feasible solution is achieved, a variable neighbourhood search heuristic (*VNS*) is used as local search.

Jans (2004) promotes new lower bounds for *CLSP*, using the network reformulation (EPPEN; MARTIN, 1987), lagrangean relaxation of demand satisfaction constraints and subgradient optimization. The resulting problem is decomposed into T (number of periods) subproblems, solvable by a branch-and-bound procedure (*BB*) whose relaxation is given by a greedy heuristic for the linear multiple choice knapsack problem. A comparison with other lower bounds of the literature is presented and the proposed lower bound performed well for a large number of iterations. The *TTM* procedure of Trigeiro *et al.* (1989) seems to be more efficient, since it provides competitive lower bounds in short computational times, presented more scalability for larger instances and also yielded good-quality feasible solutions.

Robinson & Lawrence (2004) proposed a *MILP* formulation and a branch-and-bound algorithm with lagrangean relaxation of demand satisfaction and capacity constraints for the coordinated *CLSP*. The resulting problem is easily solvable by a heuristic and presents the integrality property, *i.e.*, the lagrangean relaxation best bound is less or equal to the linear programming relaxation of the original problem. Lagrangean multipliers are found using subgradient optimization. Feasibility procedure inside a branch-and-

bound framework first restores capacity feasibility and then provides demand satisfaction feasibility. Good-quality solutions were obtained by the lagrangean heuristic solutions for the test problems, even for root node of *BB* tree, yielding 22.5% reduction in total costs compared to the industry practice.

Sambasivan & Yahya (2005) addressed the multi-plant *CLSP* with transfers, observed in a large steel industry. The approach uses lagrangean relaxation of capacity constraints, subgradient optimization and a shift production heuristic as feasibility procedure. The resulting uncapacitated problem is reformulated into a network problem, *i.e.*, a set of shortest path problems (for each plant) with common fixed-charge constraints, solved by a specialized branch-and-bound procedure.

Toledo & Armentano (2006) addressed the *CLSP* with unrelated parallel lines, using a heuristic based on lagrangean relaxation of capacity constraints and subgradient optimization. The resulting problem after the lagrangean relaxation of capacity constraints may be split in N (number of items) uncapacitated single-item subproblems, solvable by a dynamic programming algorithm. The feasibility procedure shifts production lots between lines and periods, which have their capacity increased by a factor α , temporarily, so, performing a distribution of the capacity overtime among the periods. Then, when a feasible solution is achieved, an improvement phase with lot-shifting heuristics is applied to improve the solution.

Brahimi *et al.* (2006b) proposed lagrangean relaxation based heuristics to the *CLSP* with time-windows. Two mathematical formulations were addressed: an aggregated formulation, more common in lot-sizing literature and a disaggregated model based on facility location reformulation (KRARUP; BILDE, 1977). Various strategies of lagrangean relaxation were applied, always including relaxation of capacity constraints and a combination over the time-windows constraints. Lot-shifting heuristics were proposed to turn the solution feasible, considering the relaxed constraints. The computational results suggests that the heuristic performed better when only capacity constraints were relaxed. The good quality obtained by the solutions may also suggests the application of a branch-and-bound framework to solve the problem.

Süral *et al.* (2009) considered the *CLSP* without setup costs, developing approaches based on the lagrangean relaxation of demand satisfaction constraints and the facility location reformulation of the *CLSP* (KRARUP; BILDE, 1977). The relaxed problem may be split into T (number of periods) bounded knapsack problems with setups, solvable by a specialized *BB* procedure. The lagrangean dual problem is solved by subgradient optimization. The first feasibility procedure uses the lagrangean problem with modified inventory costs in attempt to maximize the production on each period, taking into account the holding costs. Then, with the resulting setup decisions, the resulting problem is devised as a minimum cost network flow problem. The second feasibility procedure is a branch-and-bound procedure fed by a given initial solution. The results were compared

to *TTM*, with a better performance of the proposed feasibility procedure without the *BB*, however, *TTM* remains the fastest.

Cheng *et al.* (2010) developed a lagrangean relaxation and decomposition procedure for *CLSP* without setup times and multiple capacity resources (such as regular capacity and overtime), using a similar approach of Millar & Yang (1993) and Millar & Yang (1994). The difference regards the feasibility procedure, which fixes the setup plans of both decomposed subproblems in order to obtain at most two solutions using an approach analogous to Thizy & Van Wassenhove (1985). Cheng *et al.* (2013) addressed the *CLSP* with multiple resources such as Diaby *et al.* (1992b). The lagrangean relaxation is related to the capacity constraints and the lagrangean dual problem is solved by subgradient optimization. The resulting problems are split by item and solved by *WW* algorithm. Then, using the setup decisions taken, the resulting transportation subproblem is solved to achieve feasible solutions.

Fiorotto & Araujo (2014) applied lagrangean relaxation of demand constraints (flow constraints) for *CLSP* with unrelated parallel lines, using the shortest path reformulation (EPPEN; MARTIN, 1987). The resulting problem solved by the same procedure of Jans (2004), except that now the number of lines are taken into account. The lagrangean multipliers are updated by subgradient optimization. The feasibility procedure uses a heuristic which inserts, shifts and removes production to meet demand requirements.

Brahimi & Dauzère-Pérès (2014) tackled the single-item *CLSP* and variants regarding production time-windows. They proposed some properties related to the single-item case, such as a few cuts and pre-processing of the problem, generating tighter equivalent problems. A lagrangean relaxation of capacity and time-windows constraints is proposed, with lagrangean multipliers updated by subgradient optimization. The feasibility procedure shifts production, first attempting to meet time-windows requirements and then satisfying capacity constraints.

These papers are summarised in Table 4.1. As this section shows, many approaches based on lagrangean relaxation were applied in many variants of the capacitated lot-sizing problem. However, only one work has addressed *CLSP-SC* and none included perishability.

4.2 Problem statement

The *CLSP-PP* is a capacitated lot-sizing problem with deterministic and dynamic demand that should be satisfied without backlogging. Sequence-dependent setup times and costs are needed to make the line ready to process a production order. Setup carryover is assumed, *i.e.*, the setup state is maintained between adjacent periods. Perishable products have fixed shelf-life (in periods). Production orders must be assigned to periods so that spoil is avoided.

Table 4.1 – Lagrangean relaxation approaches applied to lot-sizing problems.

Reference	Problem	Constraints	Lower Bound	Feasibility procedure
Thizy & Van Wassenhove (1985)	<i>CLSP</i> without setup times	Cap	<i>DP</i> + <i>SG</i>	<i>TP</i>
Trigeiro <i>et al.</i> (1989)	<i>CLSP</i>	Cap	<i>DP</i> + <i>SG</i>	<i>SH</i>
Lozano <i>et al.</i> (1991)	<i>CLSP</i>	Cap	<i>DP</i> + Dual ascent direction	Primal-dual heuristic + <i>SH</i>
Diaby <i>et al.</i> (1992a)	<i>CLSP</i> with overtime	Dem or Cap	(<i>LP-BB</i> + <i>KP</i>) or <i>DP</i> + <i>SG</i>	<i>LR-BB</i> + <i>TP</i> at leaf nodes
Diaby <i>et al.</i> (1992b)	<i>CLSP</i> with multiple resources	Cap	<i>DP</i> + <i>SG</i>	<i>TP</i> + perturbation
Millar & Yang (1993)	<i>CLSP</i>	Lot-Size Copy	<i>TP</i> + <i>DP</i> + <i>SG</i>	<i>TP</i> with modified costs
Millar & Yang (1994)	<i>CLSP</i> with backlogging	Cap	<i>DP</i> + <i>SG</i>	<i>TP</i> with modified costs
Tempelmeier & Derstroff (1996)	Multi-level <i>CLSP</i>	Dem and Cap	<i>DP</i> + <i>SG</i>	Multi-level inventory balance + <i>SH</i>
Sox & Gao (1999)	<i>CLSP-SC</i>	Cap and single carryover	<i>DP</i> ¹ + <i>SG</i>	<i>SH</i>
Özdamar & Barbarosoglu (2000)	Multi-level <i>CLSP</i>	Dem and Cap or just Cap		<i>SH</i> + <i>SA</i>
Hindi <i>et al.</i> (2003)	<i>CLSP</i>	Cap	<i>DP</i> + <i>SG</i>	<i>SH</i> + <i>TP</i> + <i>VNS</i>
Jans (2004)	<i>CLSP</i>	Dem	<i>LP-BB</i> + <i>KP</i> + <i>SG</i>	
Robinson & Lawrence (2004)	Coordinated <i>CLSP</i>	Dem and Cap	heuristic + <i>SG</i>	<i>LR-BB</i> + production insertion heuristic
Sambasivan & Yahya (2005)	Multi-plant <i>CLSP</i> with transfers	Cap	network reformulation + <i>S-BB</i> + <i>SG</i>	<i>SH</i>
Toledo & Armentano (2006)	<i>CLSP</i> with unrelated parallel lines	Cap	<i>DP</i> + <i>SG</i>	<i>SH</i>
Brahimi <i>et al.</i> (2006b)	<i>CLSP</i> with time-windows	Cap and time-windows	<i>DP</i> + <i>SG</i>	<i>SH</i>
Süral <i>et al.</i> (2009)	<i>CLSP</i>	Dem	<i>KP</i> + <i>S-BB</i> + <i>SG</i>	Min-Cost-Flow problem + <i>S-BB</i>
Cheng <i>et al.</i> (2010)	<i>CLSP</i> with multiple resources and without setup times	Lot-Size Copy	<i>TP</i> + <i>DP</i> + <i>SG</i>	<i>TP</i>
Cheng <i>et al.</i> (2013)	<i>CLSP</i> with multiple resources	Cap	<i>DP</i> + <i>SG</i>	<i>TP</i>
Fiorotto & Araujo (2014)	<i>CLSP</i> with unrelated parallel lines	Dem	<i>LP-BB</i> + <i>KP</i> + <i>SG</i>	<i>SH</i>
Brahimi & Dauzère-Pérès (2014)	single-item <i>CLSP</i> with time-windows	Cap and time-windows	<i>DP</i> + specialized dynamic procedures + <i>SG</i>	<i>SH</i>

Cap - capacity constraints relaxation; **Dem** - demand satisfaction constraints relaxation; *SG* - subgradient optimization; *DP* - dynamic programming algorithm; *KP* - knapsack problem variant; *LP-BB* - Linear programming based branch-and-bound; *LR-BB* - lagrangean relaxation based branch-and-bound algorithm; *SBB* - specialized branch-and-bound algorithm; *TP* - transportation problem; *SH* - shift production heuristic;

¹ Procedure corrected by Briskorn (2006).

The disaggregated model of Section 3.2 is rewritten here. The formulation was chosen due to its good solution quality and tightness of linear relaxation to feasible solutions. The facility location variable reformulation proposed by Krarup & Bilde (1977) is adopted, which results in a lot-size variable that tracks the production and demand order periods. The proposed model is henceforth referred as facility location formulation (*FLF*). The indices, parameters and variables necessary to *FLF* are defined below:

Indices

i	products (items)
t, t'	periods

Parameters

N	number of items, also represent the set of items
T	number of periods, also represent the set of periods
hc_i	holding cost of item i per unit per period
sc_i	setup cost for item i
pt_i	processing time of item i per unit
st_i	setup time for item i
sl_i	shelf-life of product i (in multiples of periods)
cap_t	capacity of line in period t (in time units)
d_{it}	demand for item i in period t

Decision Variables

$X_{itt'}$	fraction of the demand satisfied for item i in period t' produced in period t
S_{it}	equals 1 if setup state i is active in period t (0 otherwise)
α_{it}	equals 1 if setup state i is active at the beginning of period t (0 otherwise)
Q_{it}	equals 1 if only setup state i is present in period t (0 otherwise)

The proposed mathematical formulation reads:

$$\text{Min } \sum_{i=1}^N \sum_{t=1}^T \sum_{t'=t}^{\min\{T, t+sl_i\}} hc_i(t' - t) d_{it'} X_{itt'} + \sum_{i=1}^N \sum_{t=1}^T sc_i(S_{it} - \alpha_{it}), \quad (4.1)$$

$$\text{s.t. } \sum_{t'=\max\{1, t-sl_i\}}^t X_{it't} = 1, \quad \forall i, t \mid d_{it} > 0, \quad (4.2)$$

$$\sum_{i=1}^N \sum_{t'=t}^{\min\{T, t+sl_i\}} pt_i d_{it'} X_{itt'} + \sum_{i=1}^N st_i(S_{it} - \alpha_{it}) \leq cap_t, \quad \forall t, \quad (4.3)$$

$$X_{itt'} \leq S_{it}, \quad \forall i, t, t' \in t, \dots, \min\{T, t + sl_i\}, \quad (4.4)$$

$$\sum_{i=1}^N \alpha_{it} \leq 1, \quad \forall t, \quad (4.5)$$

$$\alpha_{it} \leq S_{i,t-1}, \quad \forall i, t, \quad (4.6)$$

$$\alpha_{it} \leq S_{it}, \quad \forall i, t, \quad (4.7)$$

$$\alpha_{i,t+1} + \alpha_{i,t} \leq S_{it} + Q_{it}, \quad \forall i, t, \quad (4.8)$$

$$(S_{it} - \alpha_{it}) + \sum_{j=1}^N Q_{jt} \leq 1, \quad \forall i, t, \quad (4.9)$$

$$Q_{it} \leq \alpha_{it}, \quad \forall i, t, \quad (4.10)$$

$$Q_{it} \leq \alpha_{i,t+1}, \quad \forall i, t, \quad (4.11)$$

$$S_{it}, \alpha_{it} \in \{0, 1\}, \quad \forall i, t, \quad (4.12)$$

$$0 \leq X_{itt'}, Q_{it} \leq 1, \quad \forall i, t, t'. \quad (4.13)$$

The objective function (4.1) minimises the sum of holding and setup costs. Equations (4.2) ensure demand satisfaction and capacity constraints are referred to (4.3). Constraints (4.4) guarantee that the line is ready to process production order $X_{itt'}$. At most one setup state is carried over between adjacent periods (4.5). Moreover, it is necessary an active setup state in periods $t-1$ and t (4.6) and (4.7). Constraints (4.8) and (4.9) bound consecutive setup carryover variable Q_{it} . The former constraints impose that $Q_{it} = 1$ in case setup carryover variables $\alpha_{it} = \alpha_{i,t+1} = 1$. On the other hand, the latter constraints force $Q_{it} = 0$ in case production orders of different items are assigned to period t . Constraints (4.10) and (4.11) denotes the dependent relation of Q_{it} to variables α_{it} and $\alpha_{i,t+1}$. The remaining constraints state the variable domain. Although variables Q_{it} are not defined as binary, they are restricted to values 0 or 1 by the formulation.

4.3 Lagrangean heuristic

The lagrangean relaxation is applied in the facility location formulation provided in Section 4.2 ((4.1)-(4.13)). Capacity and other item-coupling constraints are relaxed to obtain N (number of products) independent uncapacitated lot-sizing problems with setup carryover. The lagrangean multipliers λ_t , μ_t and ν_{it} are associated to constraints (4.3),

(4.5) and (4.9), respectively. The lagrangean dual problem $LR(\lambda, \mu, \nu)$ provides lower bounds to the problem. A dynamic programming procedure based on Wagner & Whitin (1958) is proposed to solve $LR(\lambda, \mu, \nu)$ for an instance of lagrangean multipliers. The multipliers are updated using subgradient optimization (HELD *et al.*, 1974). A feasibility procedure is developed to search for solutions, starting from the lower bound provided by the lagrangean relaxation and is inspired on Trigeiro *et al.* (1989) lot shifting approach. The lagrangean heuristic (*LH*) is given by Algorithm 4.1 and the next sections detail each of the main steps of the heuristic.

Algorithm 4.1: Lagrangean heuristic - *LH*

Initialize lower bound (*LB*) and upper bound (*UB*) to $-\infty$ and $+\infty$, respectively;
Set iteration counter $it = 1$;
Initialize lagrangean multipliers λ^1, μ^1 and ν^1 ;
Initialize step lengths $\kappa_\lambda^1, \kappa_\mu^1$ and κ_ν^1 ;
repeat
 Solve $LR(\lambda^k, \mu^k, \nu^k)$ (Section 4.3.1);
 Calculate lower bound LB^k and update *LB* (Section 4.3.1);
 Apply feasibility procedure to the lagrangean solution achieved (Section 4.3.3);
 Update lagrangean multipliers λ^k, μ^k, ν^k and step lengths $\kappa_\lambda^k, \kappa_\mu^k$ and κ_ν^k (Section 4.3.2);
until *Stop criteria is reached*;

4.3.1 Lagrangean relaxation

The lagrangean relaxation removes some constraints of the main problem and includes them on the objective function with associated parameters (lagrangean multipliers). Capacity constraints and other item-coupling constraints are relaxed with a clear aim of separating the problem into easier single-item uncapacitated lot-sizing problems. The item-coupling constraints are given by (4.5) and (4.9), which determines, respectively, that only one setup state may be carried over from one period to the following and that the consecutive setup carryover in a period only occurs in case other setup states does not. Constraints (4.3), (4.5) and (4.9) are linked to the lagrangean multipliers λ_t, μ_t and ν_{it} , respectively. The lagrangean problem is given by (4.14).

$$\begin{aligned}
LR(\lambda, \mu, \nu) = \text{Min} \quad & \sum_{i=1}^N \sum_{t=1}^T \sum_{t'=t}^{\min\{T, t+sl_i\}} hc_i(t' - t) d_{it'} X_{itt'} + \sum_{i=1}^N \sum_{t=1}^T sc_i(S_{it} - \alpha_{it}) \\
& + \sum_{t=1}^T \lambda_t \left(\sum_{i=1}^N \sum_{t'=t}^{\min\{T, t+sl_i\}} pt_i d_{it'} X_{itt'} + \sum_{i=1}^N st_i(S_{it} - \alpha_{it}) - cap_t \right) \\
& + \sum_{t=1}^T \mu_t \left(\sum_{i=1}^N \alpha_{it} - 1 \right) + \sum_{i=1}^N \sum_{t=1}^T \nu_{it} \left((S_{it} - \alpha_{it}) + \sum_{j=1}^N Q_{jt} - 1 \right) \\
\text{s.t.} \quad & (4.2), (4.4), (4.6)-(4.8), (4.10)-(4.13).
\end{aligned} \tag{4.14}$$

The objective function (4.14) may be manipulated to be more readable as expressed in (4.15). The cost parameters of the lagrangean problem are given by (4.16) and they multiply the decision variables $X_{itt'}$ (production lot size fraction), $S_{it} - \alpha_{it}$ (setup operation),

α_{it} (setup carryover) and Q_{it} (consecutive setup carryover). The parameter *Constant* is an independent cost parameter.

$$\text{Min} \quad \sum_{i=1}^N \sum_{t=1}^T \sum_{t'=t}^T H_{itt'} X_{itt'} + \sum_{i=1}^N \sum_{t=1}^T K_{it} (S_{it} - \alpha_{it}) \\ + \sum_{i=1}^N \sum_{t=1}^T A_{it} \alpha_{it} + \sum_{i=1}^N \sum_{t=1}^T R_{it} Q_{it} - \text{Constant} \quad (4.15)$$

$$\begin{aligned} H_{itt'} &= (hc_i(t' - t) + \lambda_t p t_i) d_{it'}, \\ K_{it} &= sc_i + \lambda_t s t_i + \nu_{it}, \\ \text{where} \quad A_{it} &= \mu_t, \\ R_{it} &= \sum_{j=1}^N \nu_{jt}, \\ \text{Constant} &= \sum_{t=1}^T (\lambda_t cap_t + \mu_t + \sum_{i=1}^N \nu_{it}). \end{aligned} \quad (4.16)$$

The lagrangean problem ($LR(\lambda, \mu, \nu)$) is then split in independent subproblems, one per item. The resulting subproblem ($LR_i(\lambda, \mu, \nu)$) is an uncapacitated lot-sizing problem with setup carryover and is solved using a Wagner & Whitin (1958)'s like dynamic programming procedure. Notice that setup carryover and consecutive setup carryover operations incur on some costs in the subproblem. Moreover, as the capacity constraint is relaxed, the demand production orders are not split in multiple production lots in the optimal solution, *i.e.*, the production of a demand order occurs completely in one period ($X_{itt'} \in \{0, 1\}, \forall i, t, t'$). Therefore, variables $X_{itt'}$ are binary, although they are not explicitly defined. The subproblem of item i ($LR_i(\lambda, \mu, \nu)$) is given by (4.17)-(4.26).

$$LR_i(\lambda, \mu, \nu) = \text{Min} \quad \sum_{t=1}^T \sum_{t'=t}^T H_{itt'} X_{itt'} + \sum_{t=1}^T K_{it} (S_{it} - \alpha_{it}) \\ + \sum_{t=1}^T A_{it} \alpha_{it} + \sum_{t=1}^T R_{it} Q_{it} \quad (4.17)$$

$$\text{s.t.} \quad \sum_{t'=\max\{1, t-sl_i\}}^t X_{itt'} = 1, \quad \forall t \mid d_{it} > 0, \quad (4.18)$$

$$X_{itt'} \leq S_{it}, \quad \forall t, t' \in t, \dots, \min\{T, t + sl_i\}, \quad (4.19)$$

$$\alpha_{it} \leq S_{i,t-1}, \quad \forall t, \quad (4.20)$$

$$\alpha_{it} \leq S_{it}, \quad \forall t, \quad (4.21)$$

$$\alpha_{i,t+1} + \alpha_{i,t} \leq S_{it} + Q_{it}, \quad \forall t, \quad (4.22)$$

$$Q_{it} \leq \alpha_{it}, \quad \forall t, \quad (4.23)$$

$$Q_{it} \leq \alpha_{i,t+1}, \quad \forall t, \quad (4.24)$$

$$0 \leq X_{itt'}, Q_{it} \leq 1, \quad \forall t, t', \quad (4.25)$$

$$S_{it}, \alpha_{it} \in \{0, 1\}, \quad \forall t. \quad (4.26)$$

To solve this problem, a dynamic programming procedure (*DP*) is proposed, considering setup carryover with embedded costs. The dynamic programming procedure for each subproblem starts from period 0 and advances one period at time until period T . In the periods that some production should be made, the line should be ready to produce this order. To be ready, the line should perform a setup operation or maintain the setup state from the previous period. So, two distinct states were considered in the *DP* which denotes the operation performed in order to make the line ready to produce.

Figure 4.1 shows the *DP* for problem $LR_i(\lambda, \mu, \nu)$. Without loss of generality, all demand orders were considered positive. The nodes represent the operation performed in each period. The circled and dashed nodes of each period represent these two options in period t , respectively:

- setup carryover operation (which requires a positive setup state in the previous period $t - 1$, $S_{it} = 1$ and $\alpha_{it} = 1$);
- setup operation in current period t ($S_{it} = 1$ and $\alpha_{it} = 0$).

Moreover, the dashed nodes are named with the respective period followed by single (') prime symbols. It is crucial to track and separate these different *DP* states, because the production and setup operations of the subsequent periods depends on those previous decisions. As the method advances period by period, the best decision until period t is obtained. This decision along with the best decisions for previous periods are used to compute the best decision for the next period $t + 1$. The arcs (arrows) in Figure 4.1 give the feasible ways of meeting the demands, at the cost of the necessary operations (production, setup, setup carryover, early production and consequent holding costs). To make the graph more readable, we used the following variables to represent the cumulative costs:

$$\widetilde{H}_{itt'} = \sum_{t''=t}^{t'} H_{itt''}, \quad \widetilde{A}_{itt'} = \sum_{t''=t}^{t'} A_{itt''} \quad \text{and} \quad \widetilde{R}_{itt'} = \sum_{t''=t}^{t'} R_{itt''}.$$

The index t should be smaller or equal t' , otherwise the variables are set to zero. As the *DP* is made for each item, the subscript index i is neglected. The cost of the nodes (A_t or K_t) are assumed in all outgoing arcs. This representation of the dynamic programming procedure allow us to conclude which arcs require two consecutive setup carryover operations (dashed arcs). Consecutive setup carryover operations may be forbidden for some periods, due to the cuts detailed in Subsection 3.2.2. These cuts may impose $Q_{it} = 0$ and, as a consequence, the dashed arrows are infeasible and subsequently neglected. The arcs are also limited due to the perishable constraints, as the production can not be anticipated to some previous periods. Periods s are given for every periods from $\max\{0, t - sl_i - 1\}$ to period $t - 1$.

Figure 4.1 – DP for problem $LR_i(\lambda, \mu, \nu)$ from period 0 to period T .

In the dynamic programming procedure, each node tracks the best solution value, given by functions $f_i(t)$ and $f_i(t')$. The first node cost ($f_i(0)$) is zero. Without loss of generality, we may consider that there is a positive demand order in every period of the planning horizon. Then, demand d_{i1} must be met and the setup and production operations occur ($f_i(1) = f_i(1') = K_1 + H_{11}$) and the setup state at the end of period 1 is always present, caused by the setup operation. From period 2 and forth, the best solution values of the remaining nodes are given by Expressions (4.27). In the end, the best solution is provided by $f(T)$.

$$\begin{aligned}
 f_i(t') &= \min_{s=\max\{0, t-sl_i-1\}}^{t-1} \begin{cases} f_i(s') + \widetilde{H}_{i,s+1,t} + \widetilde{A}_{i,s+1,t} + \widetilde{R}_{i,s+1,t} ; \\ f_i(s) + \widetilde{H}_{i,s+1,t} + K_{i,s+1} + \widetilde{A}_{i,s+2,t} + \widetilde{R}_{i,s+2,t} ; \end{cases} \\
 f_i(t) &= \min_{s=\max\{0, t-sl_i-1\}}^{t-1} \begin{cases} f_i(s') + \widetilde{H}_{i,s+1,t} + A_{i,s+1} ; \\ f_i(s) + \widetilde{H}_{i,s+1,t} + K_{i,s+1} ; \end{cases}
 \end{aligned} \quad (4.27)$$

To illustrate the proposed procedure, the numerical example of Section 3.2.1 with 3 items and 4 periods is addressed here. Figure 4.2 illustrates the dynamic programming graph of the lagrangean relaxation problem of item 3 with all lagrangean multipliers equal to zero, except λ_2 , μ_2 , ν_{12} , ν_{22} and ν_{32} , all equal to 10. Node 0 has function value $f(0) = 0$, however, as there is no demand in period 1, the period does not need a setup operation if the chosen arc is $(0, 1)$ with $f(1) = 0$. In case one of the remaining arcs departing from node 0 is chosen (arcs $(0, 1')$, $(0, 2)$ and $(0, 2')$), then the node cost of 80 cost units incur. Some arcs were neglected due to perishability constraints, as the shelf-life is equal to 1, *i.e.*, products might be hold in the inventory at most 1 period. The best solution is equal to 240 cost units, highlighted by the red path in the drawing. The lagrangean multipliers in period two (mainly λ_2) makes the production in that period too expensive. Then, the production of demand order d_{32} is advanced to period 1. The remaining production operations are made in the periods correspondent to the demand order, without assuming inventory costs.

Figure 4.2 – DP for problem $LR_3(\lambda, \mu, \nu)$ from period 0 to period 4.

4.3.2 Subgradient optimization

Subgradient optimization proposed by Held *et al.* (1974) is used to solve the lagrangean dual problem, in which the best set of lagrangean multipliers is searched such that they maximize the lagrangean relaxation problem $LR(\lambda, \mu, \nu)$. The subgradient direction is given by $\Lambda_{\lambda_t}^{(k)}$, $\Lambda_{\mu_t}^{(k)}$ and $\Lambda_{\nu_{it}}^{(k)}$ in Equations (4.28). The lagrangean multipliers are first set to zero (iteration 1) and then updated for iteration $k+1$, using the subgradient directions, by Equations (4.29), (4.30) and (4.31) for λ , μ and ν , respectively. Let $z_P^{(k)}$ be the best feasible solution value achieved so far and $z_{LR}^{(k)}$ be the lagrangean relaxation dual solution value found in iteration k . Parameters $\kappa_{\lambda}^{(k)}$, $\kappa_{\mu}^{(k)}$ and $\kappa_{\nu}^{(k)}$ are the step values for iteration k . These step values are updated after a fixed number of iterations (δ) without any improvement on the lower bound, i.e., every δ iterations without improvement the step value $\kappa^{(k+1)}$ is reduced to $\kappa^{(k)} * \epsilon$, where $0 < \epsilon < 1$.

$$\begin{aligned}\Lambda_{\lambda_t}^{(k)} &= \sum_{i=1}^N \sum_{t'=t}^{\min\{T, t+sl_i\}} pt_i d_{it'} X_{itt'} + \sum_{i=1}^N st_i (S_{it} - \alpha_{it}) - cap_t \\ \Lambda_{\mu_t}^{(k)} &= \sum_{i=1}^N \alpha_{it} - 1 \\ \Lambda_{\nu_{it}}^{(k)} &= (S_{it} - \alpha_{it}) + \sum_{j=1}^N Q_{jt} - 1\end{aligned}\tag{4.28}$$

$$\lambda_t^{(k+1)} = \max \left\{ 0, \lambda_t^{(k)} + \kappa_{\lambda}^{(k)} * \frac{(z_P^{(k)} - z_{LR}^{(k)})}{\|\Lambda_{\lambda,t}^{(k)}\|^2} * \Lambda_{\lambda,t}^{(k)} \right\}\tag{4.29}$$

$$\mu_t^{(k+1)} = \max \left\{ 0, \mu_t^{(k)} + \kappa_{\mu}^{(k)} * \frac{(z_P^{(k)} - z_{LR}^{(k)})}{\|\Lambda_{\mu,t}^{(k)}\|^2} * \Lambda_{\mu,t}^{(k)} \right\}\tag{4.30}$$

$$\nu_{it}^{(k+1)} = \max \left\{ 0, \nu_{it}^{(k)} + \kappa_{\nu}^{(k)} * \frac{(z_P^{(k)} - z_{LR}^{(k)})}{\|\Lambda_{\nu_{it}}^{(k)}\|^2} * \Lambda_{\nu_{it}}^{(k)} \right\}\tag{4.31}$$

A feasible solution (or an upper bound) is necessary to the subgradient optimization procedure to update lagrangean multipliers value. However, a feasible solution may have

not been achieved yet and so a procedure of obtaining an upper bound is needed. In this case, the upper bound is provided by the most expensive function cost, where a setup cost is incurred per demand order and all orders are inventoried for the maximum time, *i.e.*, the sum of a lot-for-lot policy and the cost for all items being produced as earliest as possible, without considering setup carryover or capacity constraints. Although the obtained upper bound has a low quality, the procedure achieves the upper bound value very quickly.

4.3.3 Feasibility procedure

The feasibility procedure is applied to a solution of the lagrangean dual maximization problem, which in most of the times are not promptly feasible. Here, we have used the well known *TTM* heuristic, based on Trigeiro *et al.* (1989). However, this heuristic does not assume setup carryover and perishable products. Let a solution be setup carryover feasible if constraints (4.5) and (4.9) hold. In other words, a setup carryover feasible solution has infeasibilities caused only by period overtime. So, after the *DP* procedure, a greedy procedure turns the solution setup carryover feasible by constraining to at most one setup carryover per period, taking into account that consecutive setup carryover only occurs when there is no other setup states in that period. Then, *TTM* heuristic is applied, and a feasible solution may be achieved.

The greedy heuristic (*GCO*) receives as input a sequence of all items (Π), according to a criteria: (a) from the cheapest to the most expensive setup operation; and (b) from the shortest to the most time consuming setup operation. The setup carryover operations that will remain on the solution will be chosen according to Π . From period $t = 2$ to period T , each period becomes setup carryover feasible. First, setup carryover in period t (α_{it}) is eliminated from period $t-1$ for each item i such that $\alpha_{i,t-1} = \alpha_{it} = 1$ and another item j is processed in period $t-1$ ($S_{j,t-1} = 1$), *i.e.*, consecutive setup carryover constraints (4.9) are met. If period t has more than one setup carryover operation active, then extra operations are removed according to Π . If period t has neither setup carryover then, according to sequence Π , a setup carryover for item i may be included only if $S_{i,t-1} = S_{it} = 1$ and considering consecutive carryover constraints. Notice that including or excluding setup carryover operations, setup operations are removed or included in respective periods and therefore setup times and costs are accounted. The procedure *GCO* is always successful, since, for an extreme case, a solution without setup carryover on any period is setup carryover feasible.

The *TTM* heuristic is then applied to the setup carryover feasible solutions, regarding setup carryover and perishability features. Trigeiro *et al.* (1989) proposed this heuristic which shifts production from periods with overtime to periods with idle capacity. Here, all shifts (transfers) types maintain the solution setup carryover feasible and perishability constraints, *i.e.*, products do no spoil in inventory. The shifts are described below:

Backward pass (*BWP*): from the end of the planning horizon to the first period, each period with overtime has all its production lots evaluated to perform a shift to earlier periods. If the candidate production lot i has production and setup time smaller than the overtime of period t , complete shifts to period $t - 1$ and to the first earlier period in which a setup state to i occurs (t') are evaluated. Otherwise, in case the candidate production lot time is greater than the overtime of period t , minimal shifts to period $t - 1$ and t' (just to eliminate overtime) and a complete shift to period t' are considered. Shifts must hold cumulative capacity, *i.e.*, the setup and production capacity requirements from period 1 to t should be less than the sum of capacities from 1 to t . The evaluation of the shifts is made according to the proportion of lagrangean cost difference divided by the quantity of overtime eliminated. The best shift is performed until overtime is eliminated. Shifts may change setup carryover variables, since setup operations may be removed from periods and, in this case, another setup carryover is set in these periods.

Forward pass (*FWP*): from the first period of the planning horizon to the last period, each period with cumulative overtime (first *FWP*) or punctual overtime (second *FWP*) has all their production lots with inventory evaluated to perform a shift to the next period. Cumulative overtime means the sum of the overtime subtracted by the slack capacities from the first period to the current one and punctual overtime denotes the overtime of the current period. Only complete shifts are allowed, with all inventoried production transferred to the next period, without considering the overtime of subsequent periods. The evaluation of shifts is made according to the proportion of lagrangean cost difference divided by the quantity of capacity shifted. The best shift is performed until cumulative overtime (first *FWP*) or punctual overtime (second *FWP*) is eliminated. Again, shifts may change setup carryover variables, and other setup carryover may be stated.

Fix-up pass (*FXP*): Trigeiro *et al.* (1989) also proposed a fix-up pass, which is an improvement phase of *TTM*. The improvement procedure searches, backwardly, for periods with idle capacity and production lots with incoming inventory. This scenario offers the opportunity of processing some items in later periods to decrease holding costs. So, all candidate production shifts are evaluated in the decreasing order of the proportion of total savings per capacity shift.

The overall feasibility procedure is presented in Algorithm 4.2. Notice that if a solution reach feasibility in the first backward pass, the remaining feasibility steps are ignored, and the solution is improved using *FXP*.

4.4 Computational study

The computational tests were performed on an Intel Core i5 processor, with 2.80 GHz CPU and 8GB RAM under Linux Ubuntu 10.04 (64 bit). The lagrangean heuristic was implemented in C++. The instance sets are described in Section 3.3.1. A total of 800

Algorithm 4.2: Adapted *TTM*

Input: solution from lagrangean relaxation *sol*;
if *sol* is infeasible **then** Apply *GCO*;
if *sol* is infeasible **then** Apply first *BWP* (eliminates overtime);
if *sol* is infeasible **then** Apply first *FWP* (eliminates cumulative overtime);
if *sol* is infeasible **then** Apply second *BWP* (eliminates overtime);
if *sol* is infeasible **then** Apply second *FWP* (eliminates overtime);
if *sol* is feasible **then**
 Apply *FXP*;
 Return feasible solution;
else
 Return infeasible solution;
end

instances are generated, with five instances for each combination of (a) 12, 24, 48, 72 and 144 products; (b) 15, 30, 60 and 90 periods; (c) short, variable, medium and original shelf-life durations for perishable products; and (d) original and reduced holding costs.

The lagrangean heuristic is limited to one thousand iterations, and all of these iterations have three main phases: (a) solve the lagrangean relaxation problem using the dynamic programming procedure detailed in Section 4.3.1 to find new lower bounds to the main problem; (b) apply the proposed feasibility procedure (*ATTM*) to achieve new solutions; and (c) update the multipliers via subgradient optimization. The heuristic starts with a clear focus of turning the solution into feasibility region, and consequently, *GCO* receives as input the sequence of items on the increasing order of setup times. After achieving feasibility on the iteration, the focus turns to cost reduction (maintaining feasibility) and then the current sequence of items Π is replaced by the increasing order of setup costs. The lagrangean multipliers are first set to zero and updated according to the step size κ , initially set to 1, and after every 10 iterations without improvement on the lower bound the step size is reduced to 80% of its size.

The behaviour of four features of *LH* is analysed in the chart of Figure 4.3, considering the average of all runs. The features are, in order: (Sol) the relative difference between the best feasible solution until current iteration and the best solution found by the lagrangean heuristic; (LB) the relative difference between the best lower bound until current iteration and the best solution found by the lagrangean heuristic; (Gap) the optimality gap of the current iteration; and (Time) the computational time spent, plotted on the secondary axis. Both upper and lower bounds presented an aggressive convergence in the first iterations, smoothing in the middle of the run to the end, which is confirmed by the optimality gap curve. On average, the optimality gap is less than 3%, reaching mark after around 600 iterations. Regarding the time curve, the first iterations seem to be more time consuming than the last iterations. In a certain point, the time curve presents a linear behaviour, *i.e.*, the time consumed by the iterations is nearly equal. At the beginning of the procedure, many moves towards feasibility are needed in order to obtain a solution from a lower

bound provided by the lagrangean relaxation, as less moves are needed for the following iterations.

Figure 4.3 – Lagrangean heuristic features over the iterations.

The results obtained by *LH* are compared with the FLF60, FLF600 and FLF1800 from Chapter 3. Tables 4.2-4.5 detail the results for each of the features (Gap), (Sol), (LB) and (Time), respectively. The tables have the same structure, in which the columns represent each method and the rows denote a set of instances, depending on the parameters. The first rows represent the average and maximum of the measure gauged in the runs. Then, the remaining rows report the measurement of disjoint sets of instances defined by the distinct parameters used in the generation of the instances, such as the number of items and periods and the perishability and holding cost structures. The best results are represented in bold.

Table 4.2 shows the optimality gap (the difference of the incumbent solution value and lower bound, divided by the incumbent solution value) of the time-limited MILP-solver methods and *LH*. The proposed lagrangean heuristic presents better optimality gaps than the other methods except for instances with a smaller number of periods and higher holding costs. The results indicate that *LH* is less sensitive to the number of periods and the perishability level of the products. Besides, the method shows robustness, by achieving feasible solutions for all instances and a maximum overall optimality gap of 15.19%.

Table 4.3 reports the average of the relative difference of the incumbent solution of the method to the best solution achieved by all the methods. Let $incsol_{\alpha,\beta}$ be the incumbent solution for approach α to instance β and $bestsol_{\beta}$ the best solution achieved by all the approaches for the same instance. Therefore, the relative difference $solgap_{\alpha,\beta}$ for approach

Table 4.2 – Optimality gap of the compared methods.

	<i>FLF60</i>	<i>FLF600</i>	<i>FLF1800</i>	<i>LH</i>
Average	8.03%	4.81%	3.79%	2.83%
Maximum	68.42%	65.46%	65.45%	15.19%
Items				
12	11.95%	7.14%	6.30%	5.56%
24	10.07%	4.91%	4.20%	2.04%
48	6.17%	4.55%	3.17%	2.63%
72	5.98%	4.07%	3.18%	2.29%
144	3.64%	3.11%	1.97%	1.61%
Periods				
15	1.62%	0.92%	0.78%	2.52%
30	5.52%	3.19%	2.82%	2.59%
60	13.72%	6.13%	4.57%	3.00%
90	15.11%	9.60%	7.15%	3.20%
Perishability				
S	5.26%	3.06%	2.74%	2.49%
M	8.60%	4.51%	3.67%	2.89%
V	7.86%	3.82%	3.28%	2.90%
O	10.83%	7.91%	5.46%	3.03%
Holding Cost				
100%	2.47%	1.13%	0.87%	1.43%
25%	14.86%	8.74%	6.79%	4.22%

α and instance β is calculated by

$$solgap_{\alpha,\beta} = \frac{incsol_{\alpha,\beta} - bestsol_{\beta}}{bestsol_{\beta}}.$$

Henceforth, the measure *solgap* is referred as solution gap. The solutions of the lagrangean heuristic are better than the other methods for larger instances, considering the number of items and periods and the original problems with no perishability. The maximum solution gap is 11.61%, denoting much less variability of the solutions achieved than the MILP-solver.

Table 4.4, analogous to Table 4.3, presents the average of the measure lower bound gap: the relative difference of the incumbent lower bound of the method to the best lower bound achieved by all the methods. Let $inclb_{\alpha,\beta}$ be the incumbent lower bound for approach α to instance β and $bestlb_{\beta}$ the best lower bound achieved by all the approaches for the same instance. Therefore, lower bound gap $lbgap_{\alpha,\beta}$ for approach α and instance β is calculated by

$$lbgap_{\alpha,\beta} = \frac{bestlb_{\alpha,\beta} - inclb_{\beta}}{bestlb_{\beta}}.$$

The table aims to compare the effectiveness of the lower bound provided by the lagrangean relaxation and subgradient optimization, compared to branch-and-bound methods with linear relaxation. Table 4.4 clearly shows the better performance of such lower bound, with the worse results for smaller instances, where the MILP-solver was able to prove

Table 4.3 – Average relative difference of upper bounds for *CLSP-PP*.

	<i>FLF60</i>	<i>FLF600</i>	<i>FLF1800</i>	<i>LH</i>
Average	8.91%	3.00%	1.36%	2.13%
Maximum	143.27%	165.57%	165.57%	11.61%
Items				
12	9.17%	1.04%	0.01%	4.10%
24	10.83%	1.04%	0.04%	1.28%
48	5.74%	4.61%	1.75%	2.16%
72	9.73%	4.58%	3.04%	1.85%
144	8.76%	3.96%	1.99%	1.23%
Periods				
15	0.32%	0.03%	0.00%	2.15%
30	2.87%	0.30%	0.00%	2.20%
60	17.40%	3.33%	0.63%	2.20%
90	20.88%	9.04%	4.96%	1.95%
Perishability				
S	2.93%	0.27%	0.02%	1.92%
M	9.19%	1.86%	0.87%	2.24%
V	8.28%	0.55%	0.01%	2.16%
O	16.20%	9.43%	4.57%	2.18%
Holding Cost				
100%	1.70%	0.27%	0.01%	1.23%
25%	17.62%	5.91%	2.74%	3.03%

optimality. Therefore, the optimality gap of *LH* (Table 4.2) is better than the other methods more due to the performance of the lower bound than the feasibility procedure, indicating that efficient local search should be inserted in the lagrangean heuristic.

Finally, Table 4.5 shows computational times of the methods. The first three methods are limited to 60, 600 and 1800 seconds, respectively. The lagrangean heuristic is not limited by the computational time, though by the number of iterations. Even so, the lagrangean heuristic presented competitive computational times, with an average case of 71 seconds and a maximum computational time of around 20 minutes. The computational times of lagrangean heuristic increase as the number of items and periods increases and more computational times were spent for more expensive holding costs, the opposite of the exact methods behaviour.

4.5 Conclusion

This chapter presented a lagrangean heuristic to tackle the capacitated lot-sizing problem with setup carryover and perishable products. The developed procedure relaxes capacity constraints and other item-coupling constraints related to setup carryover and consecutive setup carryover. The lagrangean multipliers are updated by subgradient optimization. The feasibility procedure promotes the heuristic of Trigeiro *et al.* (1989), *TTM*, with some adaptations to consider perishability and the characteristics of setup

Table 4.4 – Average relative difference of lower bounds for *CLSP-PP*.

	<i>FLF60</i>	<i>FLF600</i>	<i>FLF1800</i>	<i>LH</i>
Average	3.07%	2.60%	2.50%	0.05%
Maximum	19.98%	17.95%	17.50%	2.24%
Items				
12	6.28%	5.10%	4.88%	0.21%
24	4.06%	3.55%	3.43%	0.02%
48	2.18%	2.08%	2.02%	0.01%
72	1.46%	1.48%	1.43%	0.00%
144	0.67%	0.78%	0.75%	0.01%
Periods				
15	1.03%	0.61%	0.49%	0.19%
30	2.90%	2.49%	2.38%	0.01%
60	4.07%	3.46%	3.38%	0.00%
90	4.72%	3.84%	3.76%	0.00%
Perishability				
S	2.74%	2.27%	2.17%	0.05%
M	3.10%	2.65%	2.56%	0.05%
V	3.16%	2.67%	2.56%	0.05%
O	3.33%	2.82%	2.70%	0.04%
Holding Cost				
100%	0.95%	0.72%	0.68%	0.05%
25%	5.50%	4.49%	4.33%	0.05%

Table 4.5 – Computational times for *CLSP-PP* (in seconds).

	<i>FLF60</i>	<i>FLF600</i>	<i>FLF1800</i>	<i>LH</i>
Average	58	565	1678	71
Maximum	75	676	2127	1215
Items				
12	55	517	1520	3
24	56	528	1581	6
48	60	576	1679	21
72	60	601	1800	50
144	61	602	1812	274
Periods				
15	52	459	1306	11
30	60	598	1788	26
60	60	601	1811	80
90	61	602	1808	167
Perishability				
S	58	557	1651	68
M	58	566	1673	69
V	58	568	1683	70
O	59	569	1707	76
Holding Cost				
100%	56	533	1561	84
25%	60	597	1796	58

operations.

The computational results show that the lagrangean heuristic had a better optimality gap performance, mainly because of the performance on achieving higher lower bounds. The solution values are competitive, with more robustness of the lagrangean heuristic, due to the variability of the solutions and the fact that all instances had solutions achieved by the proposed method. The computational times are also competitive. However, some improvements are needed in the feasibility procedure, such as the implementation of better local search and metaheuristic procedures. Moreover, other policies for lagrangean heuristic may be exploited such as different subgradient optimization techniques, different starts for lagrangean multipliers and different rules for applying the feasibility procedure.

5 Operational integrated production and distribution problem¹

Strategic, tactical and operational integration of the production and distribution processes is reported as being able to deliver better results for companies than a decoupled approach (PARK, 2005; AMORIM *et al.*, 2012). Very often this integration is driven by a management decision, rather than by an actual need of the underlying processes. However, when the final products are not allowed to be stocked due to, for example, freshness reasons this integration scenario becomes imperative. Within these three decision levels, it is on the operational one where more research needs to be conducted (CHEN, 2010), since actual models fail to be accurate and detailed enough for the real-world problems.

The motivation for studying the operational integrated production and distribution problem comes from very practical industry situations when it is not possible or advisable to keep final inventory decoupling these two processes. In this case, companies are forced to engage in a make-to-order production strategy. Therefore, the production for a certain demand order may only start after the order arrival. The examples found in practice are related to the computer assembly industries, the food-catering, the industrial adhesive materials or the ready-mixed concrete. The importance of a holistic vision of these processes is driven by very demanding customers requiring a product that cannot wait a long time to be delivered after production. These products, having a very short lifespan, will be called hereafter as perishable. Hence, the considered operational integrated production and distribution problem relates to the decisions on how to serve a set of customers with demand for different products. The planner has to simultaneously decide on the production planning and vehicle routing, in a setting where inventory is not allowed, *i.e.*, no inventory is carried from one planning horizon to the subsequent.

Regarding the production process, the definitions proposed by Potts & Wassenhove (1992) are followed, where batching is defined as the decision of whether or not to schedule similar jobs contiguously and lot sizing refers to the decision of when and how to split a production lot of identical items into sublots. Note that processing times are proportional to the quantities processed in both cases. The modelling of our problem considers a complex production system that is accurately synchronized with the distribution process to allow for as much flexibility as possible. Therefore, no specific industry constraints are modelled, but instead the formulation is as general as possible. Several parallel production lines with sequence dependent setups are taken into account. Moreover, the demand from different customers for a set of products has to be delivered within strict time-windows

¹ The contents of this chapter are consonants with the paper “Lot Sizing versus Batching in the Production and Distribution Planning of Perishable Goods”, referenced by (AMORIM *et al.*, 2013a).

on different routes that have to be determined together with the production planning.

So far the research community has tackled this operational integrated production and distribution problem by batching orders of customers as if lot-sizing decisions were never to yield a better solution. This is clearly not the case in the production planning literature where the importance of considering lot sizing and scheduling simultaneously is consensual for the multi-period setting (for example Almada-Lobo *et al.* (2010)). By just considering batching operations one could not achieve a production plan in which a product to a given customer is processed on different lines for example. Intuitively, however, it is observable that if the requested product is strongly perishable, then it may make sense to produce it simultaneously on both lines to ship it as soon as possible. To the best of our knowledge, the incorporation of lot-sizing decisions in the operational production and distribution problem has never been analysed. Therefore, a major contribution of this chapter is to evaluate whether lot-sizing decisions may deliver better results than batching when this integrated problem tackles perishability. After proving that lot sizing should be considered in this problem setting, the secondary contribution is to understand the conditions that improve the benefits of lot sizing versus batching.

The remainder of this chapter is organized as follows. The next section reviews the literature on the operational integrated production and distribution problem. Section 5.2 describes the considered problem and proposes two mathematical formulations for the operational production and distribution problem of perishable goods: one considering batching and the other lot sizing. In Section 5.3, the results of the computational study are presented and the impact of considering lot sizing versus batching is assessed. Finally, Section 5.4 concludes the chapter with the main findings and ideas for future work.

5.1 Literature Review

The literature in integrated production and distribution problems is vast and, therefore, only the papers very related to the scope of this work will be reviewed here. Our problem statement refers to the gap pointed out, in the review of Chen (2010), about operational integrated models dealing with *multi-customer batch delivery problems with routing*.

The research community has tackled this integrated production and distribution problem by batching orders in the production process. In Chen & Vairaktarakis (2005), orders are delivered right after their production completion time. The authors model a single product to be scheduled on the production line(s) and an unlimited number of vehicles, with a fixed capacity, which perform the routing. This work also investigates the value of integration, comparing the use of a decoupled versus an integrated approach. They conclude that the improvement is more significant when the goal is to minimize the average delivery time than the maximum delivery time. In Geismar *et al.* (2008) product

perishability is taken into account and there is a single production facility with a constant production rate. The routing process is performed by a single, capacitated vehicle that may return to the facility, therefore, performing multiple trips during the planning period. The objective is to determine the minimum makespan of the integrated production and distribution for a given set of customers. Armstrong *et al.* (2008) solve a related problem with a single product subject to a fixed lifespan that is also delivered by a single vehicle, but, in this case, there is no possibility of performing multiple trips. Moreover, the sequence of production and distribution is fixed and forced to be the same. Chen *et al.* (2009) present a model that considers stochastic demand for multiple products subject to perishability. The production environment does not consider setups between products and the delivery function is assured by a set of capacitated vehicles, however, the vehicle operating costs are disregarded. Finally, Chiang *et al.* (2009) shifts the focus to the distribution process. The production constraints influence their simulation-optimization framework through the variability of production rates and possible delays. The remaining problem is formulated as an extension to the vehicle routing problem with time-windows.

Again, none of the aforementioned papers on the operational integrated production and distribution planning include lot-sizing decisions. However, on pure production scheduling, the advantages of lot sizing over batching for a leaner environment have been proven. Santos & Magazine (1985), Wagner & Ragatz (1994), Low & Yeh (2008) show how lot sizing can reduce lead time in the scheduling of machines and the impact of setup times is investigated. Nieuwenhuyse & Vandaele (2006) proves that lot sizing improves the reliability of the deliveries in a system accounting for production and direct deliveries to customers. Moreover, in make-to-order environments with a multi-level production structure, Anwar & Nagi (1997) show the advantages of lot sizing compared against a lot-for-lot strategy. The scope of these related papers, however, does not include the distribution decision carried in the present work.

Based on this literature review the contribution of this study is clearer. Firstly, it investigates the potential performance improvement that lot-sizing decisions may add to the operational production and distribution planning (in relation to only batching orders). Secondly, previous studies are extended by considering a more general production system with sequence-dependent costs and times between products.

5.2 Problem Statement and Mathematical Formulations

In this section, the problem statement is given as well as two mathematical formulations for this problem. The first formulation models the operational integrated production and distribution problem that only considers batching of orders (I-BS-VRPTW) and the second formulation extends the first one by considering the sizing of the lots (I-LS-VRPTW). Both models are then compared.

The operational integrated production and distribution planning problem considered in this work consists of a set M of parallel lines $l = 1, \dots, m$ with limited capacity that produce a set P of items (or products) $i, j = 1, \dots, p$ to be delivered to a set N of customers $c, d = 1, \dots, n$ through a set A of arcs (c, d) . The delivery is assured by a set K of identical fixed capacity vehicles indexed by $k = 1, \dots, o$ initially located at a depot. Hence, the routing can be defined on a directed graph $G = (V, A)$, $V = N \cup \{0, n + 1\}$, where the depot is simultaneously represented by the two vertices 0 and $n + 1$, and, therefore, $|V| = n + 2$.

Some of the products may be perishable while others last substantially beyond the considered planning horizon. Furthermore, the utilization of equipment, such as ovens in the food-catering, makes the changeover between different products dependent on the sequence. Hence, products are to be scheduled on the parallel production lines over a finite planning horizon that ranges up to the time of the last scheduled delivery.

The distribution is performed using several vehicles serving multiple customers on different routes. There exists a variable cost dependent on the total distance travelled and a fixed cost for each vehicle used. It is assumed that there are no fleet constraints such that any distribution plan can be executed. This assumption is realistic since reference contracts are usually established assuring that there always exists a fleet of sufficient size available. The two models determine the routing taking into account the vehicle capacity, and the time and cost to travel from one customer/depot to another. A customer order may aggregate several products that have to be delivered within strict time-windows with a single delivery (i.e., split deliveries are not allowed). Moreover, it is assumed that demand is deterministic.

The challenge is to model the production and distribution problem that minimizes total cost of the supply chain covering these processes over the short planning horizon.

The main advantage of these models comes from the accurate synchronization of the two planning processes. While at the tactical level, the integrated production and distribution planning has the possibility to assume that at the end of the period, after production, one will start the delivery process to all customers, this assumption is not possible at the operational level. At this level one needs to go one step further and be sure that the production times of the customer orders are accurately traced so that as soon as a customer has his order completed, the vehicle servicing him may depart. However, the departure only takes place after the last customer's order (serviced by the same vehicle) has been produced.

Consider the following indices, parameters and decisions variables that are needed to formulate both the I-BS-VRPTW and the I-LS-VRPTW models. Notice that the variable names are different from the nomenclature utilised on the previous chapters.

Parameters

dem_{jc}	demand for product j at customer c (units)
$cpl_j(tp_{lj})$	production cost (time) of product j (per unit) on line l
$scb_{lij}(stb_{lij})$	sequence dependent setup cost (time) of a changeover from product i to product j on line l
α_l	initial product set up on line l
sl_j	shelf-life of product j (time)
Cap_l	available capacity (= latest completion time) of production line l
$CapV$	vehicle capacity on each trip
s_c	service time of customer c
$ct_{cd}(tt_{cd})$	cost (time) of travelling from customer c to d
$\bar{f}t$	fixed cost associated with each vehicle k
$[a_c, b_c]$	time-window for customer c

Decision Variables

f_c	completion time of the production of customer c 's order
x_{cd}^k	equals 1, if arc (c, d) is used by vehicle k (0 otherwise)
w_c^k	starting time at which vertex c is serviced by vehicle k

5.2.1 Integrated Batch Scheduling and Vehicle Routing Problem (I-BS-VRPTW)

This production planning modelling of this formulation is based on the work of Mendez *et al.* (2000). A job is given by each pair product-customer (j, c) with positive demand. Let H denote the set of these jobs ($H = \{(j, c), j \in P, c \in N \mid dem_{jc} > 0\}$).

In order to formulate the integrated problem considering batching decisions, the following additional decision variables are needed to be added to the aforementioned ones.

Decision Variables

$R_{l(j,c)}$	equals 1, if job (j, c) is produced on line l (0 otherwise)
$RO_{l(j,c)}$	equals 1, if job (j, c) is the first to be produced on line l (0 otherwise)
$RN_{l(j,c)}$	equals 1, if job (j, c) is the last to be produced on line l (0 otherwise)
$V_{l(i,d)(j,c)}$	equals 1, if job (j, c) is scheduled right after (i, d) on line l (0 otherwise)
$Ct_{(j,c)}$	completion time of job (j, c)

The batch scheduling coupled with the vehicle routing problem with time-windows (I-BS-VRPTW) may be formulated as follows:

I-BS-VRPTW

$$\text{Min} \quad \sum_{l,(i,d),(j,c)} scb_{lij} V_{l(i,d)(j,c)} + \sum_{l,(j,c)} scb_{l\alpha_l,j} R_{0l(j,c)} + \sum_{l,(j,c)} cpl_{(j,c)} dem_{jc} R_{l(j,c)} + \bar{f}t \sum_k (1 - x_{0,n+1}^k) + \sum_k \sum_{c,d} ct_{cd} x_{cd}^k \quad (5.1)$$

$$\text{s.t.} \quad \sum_{(j,c)} R_{0l(j,c)} \leq 1, \quad \forall l, \quad (5.2)$$

$$R_{0l(j,c)} \leq R_{l(j,c)}, \quad \forall l, (j, c), \quad (5.3)$$

$$\sum_{(j,c)} R_{Nl(j,c)} \leq 1, \quad \forall l, \quad (5.4)$$

$$R_{Nl(j,c)} \leq R_{l(j,c)}, \quad \forall l, (j, c), \quad (5.5)$$

$$\sum_l R_{l(j,c)} = 1, \quad \forall (j, c), \quad (5.6)$$

$$R_{l(i,d)} + V_{l(i,d)(j,c)} \leq R_{l(j,c)} + 1, \quad \forall l, (i, d), (j, c), \quad (5.7)$$

$$\sum_l R_{0l(j,c)} + \sum_{(l,i,d)} V_{l(i,d)(j,c)} = 1, \quad \forall (j, c), \quad (5.8)$$

$$\sum_l R_{Nl(j,c)} + \sum_{(l,i,d)} V_{l(j,c)(i,d)} = 1, \quad \forall (j, c), \quad (5.9)$$

$$Ct_{(j,c)} \geq Ct_{(i,d)} + \max_l \{Cap_l\} (\sum_l V_{l(i,d)(j,c)} - 1) + \sum_l (tp_{lj} dem_{jc} + stb_{lij}) R_{l(j,c)}, \quad \forall (i, d), (j, c), \quad (5.10)$$

$$Ct_{(j,c)} \geq \sum_l (tp_{lj} dem_{jc} + stb_{l\alpha_l,j}) R_{0l(j,c)}, \quad \forall (j, c), \quad (5.11)$$

$$Ct_{(j,c)} \leq \max_l \{Cap_l\} + (Cap_l - \max_l \{Cap_l\}) R_{l(j,c)}, \quad \forall l, (j, c), \quad (5.12)$$

$$f_c \geq Ct_{(j,c)}, \quad \forall (j, c), \quad (5.13)$$

$$Ct_{(j,c)} - tp_{lj} dem_{jc} + sl_j - \sum_k w_c^k \geq 0, \quad \forall l, (j, c), \quad (5.14)$$

$$w_0^k \geq f_c - \max_l \{Cap_l\} (1 - \sum_d x_{cd}^k), \quad \forall k, c, \quad (5.15)$$

$$\sum_k \sum_d x_{cd}^k = 1, \quad \forall c, \quad (5.16)$$

$$\sum_d x_{0d}^k = 1, \quad \forall k, \quad (5.17)$$

$$\sum_c x_{cd}^k - \sum_c x_{dc}^k = 0, \quad \forall k, d, \quad (5.18)$$

$$\sum_c x_{c,n+1}^k = 1, \quad \forall k, \quad (5.19)$$

$$w_d^k \geq w_c^k + s_c + tt_{cd} - \max_l \{Cap_l\}(1 - x_{cd}^k), \quad \forall k, c, d, \quad (5.20)$$

$$a_c \sum_d x_{cd}^k \leq w_c^k \leq b_c \sum_d x_{cd}^k, \quad \forall k, c, \quad (5.21)$$

$$\sum_{(j,c)} dem_{jc} \sum_d x_{cd}^k \leq CapV, \quad \forall k, c, \quad (5.22)$$

$$f_c, Ct_h, w_c^k \geq 0, \quad (5.23)$$

$$R_{lh}, R0_{lh}, RN_{lh}, V_{h'h}, x_{cd}^k \in \{0, 1\}.$$

The main supply chain related costs are minimized with objective function (5.1). The first terms relate to sequence dependent setup costs. The second term is used to trace the first setup incurred. Variable productions costs are considered in the third term and the last two terms are related to the distribution costs, namely fixed costs for each vehicle used and variable transportation costs.

Constraints (5.2) - (5.6) assign each job (j, c) to a line either in the beginning, in the end or in the middle of the scheduling sequence. Constraints (5.7) ensure that consecutive jobs are assigned to the same line. Equations (5.8) establish that a job is either assigned in the beginning of the scheduling or preceded by other job. Similarly, equations (5.9) impose that a job is assigned at the end of the scheduling or precedes other job. For tracing the completion time of each job, constraints (5.10) and (5.11) are used. Note that in (5.10), $\max_l \{Cap_l\}$ denotes the latest possible completion time due to the capacity limitations of the lines. Also, these constraints are responsible for the job scheduling. Job completion time must not exceed the available capacity of the line which is assigned (5.12). Constraints (5.13) define f_c , which tracks the customer order finishing time. To account for perishability, (5.14) assures that the delivery is performed while products still have some lifetime.

In (5.15) the link between production and the vehicle departing times is established. This synchronization ensures that a vehicle only departs after the completion of the production for all customers visited along the vehicle's route. Constraints (5.16)-(5.22) refer to the distribution process. Each customer is visited exactly once by (5.16), while constraints (5.17)-(5.19) ensure that each vehicle is used once and that flow conservation is

satisfied at each customer vertex. $x_{0,n+1}^k = 1$ means that the vehicle was not used. The consistency of the time variables w_c^k is ensured through constraints (5.20), while time-windows are imposed by (5.21). Regarding the vehicle capacity, constraints (5.22) enforce it to be respected. Finally, the domain of the variables is limited by (5.23).

5.2.2 Integrated Lot Sizing and Scheduling and Vehicle Routing Problem (I-LS-VRPTW)

Due to production planning modelling reasons, the planning horizon is divided in the lot-sizing formulation into a fixed number of non-overlapping slots, indexed by s , of variable length. Since the production lines can be independently scheduled, this partition is done for each line separately ($s \in S_l$). The length of a production slot is a decision variable that is a function of the production quantity of a certain product on a line and of the time to set up the machine for this product (in case it is required). A sequence of consecutive production slots, where the same product is produced on the same line, defines the size of the lot of a product. Therefore, a lot may span over several slots. The number of production slots of a certain line defines the upper bound on the number of setups and deliveries to be performed during the planning horizon.

Contrarily to the more tactical lot-sizing and scheduling formulations that integrate the delivery process (BOUDIA *et al.*, 2007), this model considers a continuous time scale since the external factors, such as demand are pulled from the customer desires, expressed in its time-window boundaries. Notice that the slot structure of the mathematical formulation related to the production planning resembles the micro-period time structure of the general lot-sizing and scheduling problem (FLEISCHMANN; MEYR, 1997).

Consider the additional decision variables.

Decision Variables

q_{ljs}^c	quantity of product j produced in slot s on line l to serve customer c
y_{ljs}	equals 1, if line l is set up for product j in slot s (0 otherwise)
z_{ljs}	equals 1, if a changeover from product i to product j takes place at the beginning of slot s on line l (0 otherwise)
str_{ls}	starting time of production slot s on line l
λ_{ljs}^c	equals 1, if there is production of product j for customer c in production slot s on line l (0 otherwise)
F_j^c	starting time of the lifespan of product j for customer c

The lot-sizing and scheduling coupled with the vehicle routing problem with time-windows (I-LS-VRPTW) is formulated as follows:

I-LS-VRPTW

$$\text{Min } \sum_{l,i,j,s} scb_{lij} z_{lij} + \sum_{l,j,s,c} cpl_j q_{ljs}^c + \bar{f}t \sum_k (1 - x_{0,n+1}^k) + \sum_k \sum_{c,d} ct_{cd} x_{cd}^k, \quad (5.24)$$

$$\text{s.t. } \sum_{l,s} q_{ljs}^c = dem_{jc}, \quad \forall j, c, \quad (5.25)$$

$$\sum_c q_{ljs}^c \leq \frac{Cap_l}{tp_{lj}} y_{ljs}, \quad \forall l, j, s, \quad (5.26)$$

$$\sum_j y_{ljs} = 1, \quad \forall l, s, \quad (5.27)$$

$$y_{l\alpha_l,0} = 1, \quad \forall l, \quad (5.28)$$

$$\sum_{i,j,s} stb_{lij} z_{lij} + \sum_{j,s,c} tp_{lj} q_{ljs}^c \leq Cap_l, \quad \forall l, \quad (5.29)$$

$$z_{lij} \geq y_{li,s-1} + y_{ljs} - 1, \quad \forall l, i, j, s, \quad (5.30)$$

$$str_{ls} \geq str_{l,s-1} + \sum_{i,j} stb_{lij} z_{lij,s-1} + \sum_{j,c} tp_{lj} q_{ljs}^c, \quad \forall l, s > 1, \quad (5.31)$$

$$q_{ljs}^c \leq dem_{jc} \lambda_{ljs}^c, \quad \forall l, j, s, c, \quad (5.32)$$

$$f_c \geq -Cap_l (1 - \sum_j \lambda_{ljs}^c) + str_{ls} + \sum_{i,j} stb_{lij} z_{lij} + \sum_{j,d} tp_{lj} q_{ljs}^d, \quad \forall l, s, c, \quad (5.33)$$

$$F_j^c \leq Cap_l (1 - \lambda_{ljs}^c) + str_{ls} + \sum_i stb_{lij} z_{lij}, \quad \forall l, j, s, c, \quad (5.34)$$

$$F_j^c + sl_j - \sum_k w_c^k \geq 0, \quad \forall j, c \mid dem_{jc} > 0, \quad (5.35)$$

$$(5.15) - (5.22),$$

$$q_{ljs}^c, z_{lij}, str_{ls}, f_c, F_j^c, w_c^k \geq 0, \quad (5.36)$$

$$y_{ljs}, \lambda_{ljs}^c, x_{cd}^k \in \{0, 1\}.$$

In the objective function (5.24) the same costs are minimized as in the batching related formulation. In this case the first term is enough to account for the sequence dependent setups related costs.

Looking now at the constraints that this problem is subject to, demand is to be satisfied with production that may come from different lines (5.25). Constraints (5.26) ensure that a product can only be produced if there exists a setup for it and constraints (5.27) limit to one the number of products to be simultaneously produced on each line. Constraints (5.28) set the initial configuration of the lines. Limited capacity in the lines is to be used by setup times and the time consumed producing products (5.29). The connection between setup states and changeover indicators for products is established by (5.30). In order to define f_c that tracks the customer order finishing time in constraint (5.33), the starting time of each production slot is traced with (5.31). Requirements (5.32) determine the customers for which the production in a given slot is devoted to. It is worth mentioning that this production may satisfy demand from several customers. Constraints (5.34) and (5.35) account for product perishability similarly to equations (5.13) and (5.14). Note, that the model formulation allows for the production of the same product for different customers in a single slot. In such case, f_c and F_j^c are considering only the end and the start of the time slot, therefore, these variables are not considering the exact time of production for each customer. However, this situation can always be avoided by producing the same product of different customer orders in separate (possibly subsequent) slots without additional cost or capacity needs ($sch_{lji} = stb_{lji} = 0$). Constraints (5.15)-(5.22) from the previous model are also used in this one. The domain of variables is stated in (5.36) and the remaining constraints come from the integrated model with batching decisions (I-BS-VRPTW).

5.2.3 Relation Between both Models

The meaning of the main decision variables of both formulations is graphically presented in Figure 5.1. It is easy to see that both solutions of this illustrative example are equivalent, as the two jobs of I-BS-VRPTW are not split in the I-LS-VRPTW. While the production quantities need to be explicitly tracked in the I-LS-VRPTW, this is not the case for the batching model. In terms of setup variables, the two models are very similar, but the lot-sizing model has these variables linked to the micro-period, whereas the I-BS-VRPTW uses a continuous representation. The vehicle routing problem with time-windows is known to be NP-hard (SAVELSBERGH, 1985). Consider special cases of the I-LS-VRPTW and I-BS-VRPTW where all the products have shelf-lives (sl_j) equal to $+\infty$ and null processing times ($tp_{lj} = 0$, for every product j on line l). Furthermore, let the setup costs and times be equal to zero (sch_{lij} e stb_{lij} , for every l, i, j). Then, the VRP with time-windows can be solved in polynomial time if this instance of the I-LS-VRPTW and I-BS-VRPTW can be solved in polynomial time. Therefore, we can conclude that both I-LS-VRPTW and I-BS-VRPTW are NP-hard.

In the following theorem it is shown that the optimal solution to I-LS-VRPTW is at least as good as the optimal solution to I-BS-VRPTW. Let $\nu(\cdot)$ denote the optimal values of underlying optimization problems.

Figure 5.1 – Comparing the decision variables of I-BS-VRPTW and I-LS-VRPTW.

Theorem 5.1. *We have $\nu(I - LS - VRPTW) \leq \nu(I - BS - VRPTW)$.*

Proof. We prove the statement by showing that I-BS-VRPTW is a special case of I-LS-VRPTW and therefore any feasible solution to I-BS-VRPTW is also feasible to I-LS-VRPTW. Let model $\bar{f}_{LS}$ be derived from I-LS-VRPTW by adding to the latter the following constraints: $\sum_{l,s \in S_l} \lambda_{ljs}^c = 1$, for every j in N and c in C , and $\sum_{c,j} \lambda_{ljs}^c = 1$, for every l and s in S_l . These conditions mean that demand for a given pair product j -customer c is produced in just one lot, and that each production slot can only be allocated to pair $j - c$. Now, we show the equivalence between I-BS-VRPTW and $\bar{f}_{LS}$. Let $Q^*(f_c, Ct_h, w_c^k, R_{lh}, R0_{lh}, RN_{lh}, V_{h'h}, x_{cd}^k)$ be a feasible solution to I-BS-VRPTW. Each job h entails a product j to be produced and delivered to a customer c . Consider in the following a given line l . Each job of I-BS-VRPTW relates to one production slot of $\bar{f}_{LS}$. The sequence $(h_1, h_2, \dots, h_g)$ can be easily transformed into the sequence $(j_1 - c_1, j_1 - c_2, \dots, j_p - c_n)$, where the quantity of each product produced in each slot (q_{ljs}^c) equals the amount of demand of the respective job. In case job h in I-BS-VRPTW is produced in the s -th position of the sequence, its completion time (C_h) is equivalent in $\bar{f}_{LS}$ to the finishing time of the s -th slot where the respective product j is produced to supply the same customer c (i.e. $C_h = str_{ls} + \sum_{i,j} stb_{lij} z_{lij} + \sum_{j,c} tp_{ij} q_{ljs}^c$). Moreover, the starting time of the lifespan of product j for customer c (F_j^c) in $\bar{f}_{LS}$ is equivalent to the term $C_h - tp_{lh}$ of the respective job in I-BS-VRPTW. Clearly, Q fulfils the constraints related to the production part of $\bar{f}_{LS}$, from (5.25) to (5.35). The routing-related requirements are the same in both formulations. This clearly shows that $\nu(I-LS-VRPTW) \leq \nu(\bar{f}_{LS}) \leq \nu(I-BS-VRPTW)$. $\square$

5.3 Computational Study

This section aims at quantifying the impact of considering lot sizing versus batching and analysing the solution changes that this extra production flexibility yields. To this end, a set of instances have been systematically generated with different parameters. Next it is reported how the test instances are generated. Afterwards, the computational results are presented and, finally, some examples comparing the improvements of the lot sizing over the batching solutions are analysed.

5.3.1 Data Generation

The instance generators used by Haase & Kimms (2000), Armstrong *et al.* (2008) and Viegut (2011) are integrated since, to the best of our knowledge, there are no reported instances for the settings of this problem. A total of 120 instances were generated. The impact of different key parameters on the lot sizing importance is verified by varying: the *number of perishable products*, the *length of the shelf-life*, the *setup time and cost structure*, the *tightness of the time-windows* and the *orientation of the time-windows*.

For the sake of compactness, the description of parameters' generation is exposed only for I-LS-VRPTW. However, the data conversion for I-BS-VRPTW is straightforward. The number of lines m is set to 1 and for all products $tp_{lj} = 1$ and $cp_{lj} = 0$. In the beginning of the planning horizon the machine is set up for product 1. There are 3 items ($p = 3$) to be produced for 5 customers ($n = 5$). The number of production slots S_l is set to $p \times n$ in order to ensure that all necessary setups and deliveries may be performed. 75% of the demand dem_{jc} is generated from the uniform distribution in the interval $U[40, 60]$ and the remaining 25% is set to 0 Trigeiro *et al.* (1989). The *number of perishable products* (PP) can be 1 or 2 out of 3 items. In order to define the *length of the shelf-life* of perishable products (sl_j), parameter SL is multiplied by the average production time of a demand order. This parameter SL can be 3 or 5, where 3 corresponds to highly perishable products and 5 to average perishable products.

The *setup time and cost structure* may obey or not to the triangular inequality. In case setups obey to triangular inequality, in the optimal solution the production of the same product will never take place twice in the same period. On the contrary, setups not obeying to the triangular inequality, which are frequent in the food industry with the use of cleaning products, may result in optimal solutions in which the same product is set up more than once in the same period (favouring the lot sizing). For the instances with triangular setup times (TS) between products stb_{lij} , $U[6, 10]$ is used for all pairs (except for the case where $i = j$, where the setup is 0). The instances not obeying to such inequality (NTS) have setup times chosen randomly from $U[1, 5]$. The setup costs scb_{lij} of a changeover from product i to j are computed as:

$$scb_{lij} = 25.0 \cdot stb_{lij} \text{ and } scb_{lij} = 66.67 \cdot stb_{lij},$$

for triangular and non-triangular setups, respectively. Combining the distribution of the setup times and the coefficients of 25.0 and 66.67 we assure that instances with either triangular or non-triangular setup structures have an average setup cost of 200 units. The line capacity Cap_l is determined according to:

$$Cap_l = \frac{\sum_{jc} dem_{jc} t_{plj}}{0.6}.$$

With this expression, based on Haase & Kimms (2000), we define the capacity utilization to be around 60%. This is an estimate only, as setup times do not influence the computation of Cap_l (just the variable production time). We also enforce minimum batch sizes to be equal to the smaller order quantity, therefore, this condition only influences the lot-sizing model in order not to perform very small production lots.

For the computation of the travel times tt_{cd} and costs ct_{cd} , which are assumed to be the same, all customers are positioned randomly in a square of locations from (0,0) to (100,100). The Euclidean distance is then calculated between all pairs of customers (assuming that travel times are equal to the travel distances) fixing the depot at the point (50,50). The number of available vehicles is set to n (number of customers) and the cost of using each vehicle $\bar{f}t$ is set to 250. This value was set after preliminary computational experiments to reflect the industry practice in relation to the vehicle variable costs. The capacity of the vehicle is computed through the expression

$$CapV = 0.5 \sum_{jc} dem_{jc}.$$

For all customers, it is considered that no service time (s_c) is necessary. The last parameters are the time-windows of each customer (parameters a_c and b_c), which are calculated by four different methods that change the *tightness* and the *orientation of the time-windows*. With regard to the *tightness*, instances with standard (S) and loose (L) time-windows are considered. Concerning their *orientation*, instances with time-windows oriented by production requirements (P) and by customers' demand (C) are assumed.

For the generation of time-windows data, an auxiliary parameter τ (that estimates the length of a vehicle tour) needs to be defined in two steps. First, a greedy nearest neighbourhood procedure finds a path for all customers without considering time-windows. The distance of the solution found is then multiplied by 0.5 in order to account for the necessary expected vehicles (recall that a vehicle is able to carry half of the total demand), defining τ . Let us now define μ_{tw} as the mean width of the time-windows that equals to 0.1τ . Moreover, be the maximum setup time denoted by $maxStb$ and the value of the average demand element by $avDem$. Two different methods are responsible for varying the *orientation of time-windows*: production (P) or customer (C) oriented. To generate these time-windows the algorithm proposed in Viegutz (2011) was adapted and described.

In Algorithm 5.1, the generated customer's time-windows are production (P) oriented, in the sense that the first time-window just starts after the necessary time to complete

half of the total demand. Variables $auxWidth$ and $auxGap$ state the maximum width of a time-window and the maximum gap between two consecutive start times, respectively. Then, the width of the time-window and the gap between two consecutive start times are randomly generated. Algorithm 5.2 describes the generator of customer (C) oriented time-windows and yields a profile in which parameters a_c and b_c are now defined according to the demand of each customer. The time-window of a customer c initiates after taking into account the sum of the processing times of demand orders and setup times of all customers before c , including itself. Moreover, the travel time from the depot to customer c is considered. The time-window is then generated, depending on μ_{tw} and $avDem$.

Algorithm 5.1: Pseudo-code to generate production (P) oriented time-windows

```

 $aux \leftarrow 0.5 \sum_{jc} dem_{jc};$ 
 $auxWidth \leftarrow 2/5\mu_{tw};$ 
 $auxGap \leftarrow 2/5avDem;$ 
for  $c = 1 \rightarrow n$  do
     $a_c \leftarrow aux;$ 
     $auxLow \leftarrow \max\{0, \mu_{tw} - auxWidth/2\};$ 
     $twWidth \leftarrow RANDOM(auxLow, auxLow + auxWidth);$ 
     $b_c \leftarrow a_c + twWidth;$ 
     $auxLow \leftarrow \max\{1, avDem - auxGap/2\};$ 
     $Gap \leftarrow RANDOM(auxLow, auxLow + auxGap);$ 
     $aux \leftarrow aux + Gap;$ 
end

```

Algorithm 5.2: Pseudo-code to generate customer (C) oriented time-windows

```

 $aux = 0;$ 
 $auxWidth \leftarrow 2/5\mu_{tw};$ 
for  $c = 1 \rightarrow n$  do
    for  $j = 1 \rightarrow p$  do
        if  $d_{jc} > 0$  then  $aux \leftarrow aux + d_{jc} + maxStb;$ 
    end
     $auxLow \leftarrow \max\{0, \mu_{tw} - auxWidth/2\};$ 
     $a_c \leftarrow aux + tt_{0c} - auxLow;$ 
     $b_c \leftarrow a_c + auxLow + avDem/2;$ 
end

```

In order to vary the *tightness of time-windows*, the standard (S) tightness of the time-windows calculated in Algorithms 5.1 and 5.2 is relaxed to achieve loose (L) time-windows. Hence, the L time-windows are calculated by postponing by 20% the time-windows calculated with the previous two methods, *i.e.*, time-window values a_c and b_c are multiplied by 1.2. Consequently four different types of time-windows may be generated, considering the tightness and the orientation of the time-windows.

By varying the aforementioned parameters, 24 types of instances are generated. For each of them, 5 random instances are considered. All the 120 instances were tested

for feasibility purposes on the I-BS-VRPTW model with a commercial solver. In case a solution had not be found, then a new instance was generated until feasibility was achieved.

5.3.2 Computational Results

All computational experiments were performed on an HP Z800 workstation with two six-core Intel Xeon X5690 at 3.47 GHz with 48 GB RAM, running Linux. CPLEX version 12.4 from IBM was used as the MIP solver. The data generator described in Section 5.3.1 was used to obtain the instance set. The computational time to solve each MIP is limited to 3600 seconds. As the I-BS-VRPTW was solved to optimality by CPLEX within a maximum/average running time of 126.97/6.07 seconds, these solutions were used as a starting point for the I-LS-VRPTW (*i.e.* they were injected into its branch-and-bound tree).

In order to evaluate the solutions of the two models we use indicator gap_{sol} , that refers to the relative difference of solutions between the I-LS-VRPTW (UB_L) and I-BS-VRPTW (UB_B). These gap is calculated as:

$$gap_{sol} = \frac{UB_B - UB_L}{UB_L}.$$

Table 5.1 provides the solution improvement gap_{sol} of I-LS-VRPTW over I-BS-VRPTW for the all the instances. Here, The sign “-” means that the I-BS-VRP-TW solution was not improved by I-LS-VRP-TW model, within the time limit. Notice that the average integrality gap of the lot-sizing solutions is 6.3%. The cause behind the solution improvements is also presented in the same table. In general, the cost decrease on the solution of I-LS-VRPTW may yield five main changes in relation to the solution of I-BS-VRPTW:

- St-(+): number of setup operations;
- Sc-(+): total setup cost;
- Seq: setup sequence;
- Dist-(+): distance travelled;
- V-(+): number of used vehicles.

The signs - (+) mean a decrease (increase) of the indicator of the respective change. Notice that, contrarily to the case of triangular setup structure, the case of non-triangular setups may allow for setup inclusions (St+) that result in setup cost reduction (Sc-). Therefore, Sc- is omitted for triangular setups when the related changes are due to St- or Seq.

Table 5.2 contains the detailed absolute costs of both formulations for all instances. The total cost is composed of setup, vehicle and travel costs. In case the solution values are different a slash is used to distinguish both values (I-BS-VRPTW/I-LS-VRPTW).

I-LS-VRPTW obtained better solutions for 35 out of 120 instances. In 22 instances both formulations reported the same provably optimal solution and there are 40 instances for which it is still theoretical possible to improve the batching solution by allowing for lot size. The maximum gap_{sol} is 20.0% caused by the reduction of setup operations. The average gap_{sol} , for instances with positive gaps, is 6.5%. The main cause of cost decrease, when lot sizing is allowed, is the reduction of setup operations, which was responsible for 21 out of the 35 instances improved. The advantage of the I-LS-VRPTW formulation is clear for instances with customer oriented time-windows (C) and non-triangular setups (NTS). Moreover, loose time-windows (L) allowed more changes related to distribution decisions. Regarding the perishability phenomenon, results indicate that smaller shelf-lives ($SL = 3$) also augment the benefits of using the I-LS-VRPTW formulation. The usage of the I-BS-VRPTW formulation could be justifiable when having as conditions standard product oriented time-windows and a triangle setup structure ($P - S - TS$).

Table 5.1 – Gaps between batching and lot-sizing solutions.

PP	SL	#	P-S-TS	P-L-TS	P-S-NTS	C-S-TS	C-L-TS	C-S-NTS
1	3	1	-	-	-	2.9% (St-)	-	6.1% (Seq)
		2	-	-	-	11.2% (St-)	1.7% (Dist-)	1.3% (Dist-)
		3	-	-	-	-	-	-
		4	-	3.6% (V-,Dist-,St+)	-	15.3% (St-)	8.7% (St-)	9.0% (Seq)
		5	-	3.4% (Dist+,St-)	6.8% (St-)	2.7% (St-)	8.7% (Dist+,St-)	-
	5	1	-	-	-	1.0% (Seq)	-	-
		2	-	-	-	-	-	2.9% (St+,Sc-)
		3	-	6.3% (V-,Dist-,St+)	-	-	2.3% (St-)	6.0% (St-)
		4	-	-	-	-	-	-
		5	-	-	3.9% (Dist+,St-)	-	-	-
	2	1	9.3% (St-)	-	15.3% (Seq)	8.1% (St-)	9.3% (St-)	2.4% (St+,Sc-)
		2	-	-	-	-	-	-
		3	-	-	-	13.3% (St-)	-	20.0% (St-)
		4	-	-	2.7% (St+,Sc-)	-	-	-
		5	-	-	-	16.3% (St-)	-	2.4% (St+,Sc-)
2	3	1	-	-	9.1% (Dist+,St-)	-	2.5% (St-)	0.9% (Dist-)
		2	-	-	3.4% (St-)	-	-	-
		3	-	-	-	-	-	-
		4	-	-	-	-	-	4.9% (St-)
		5	-	-	-	-	-	3.4% (St+,Sc-)
	5	1	-	-	-	-	-	-
		2	-	-	-	-	-	-
		3	-	-	-	-	-	-
		4	-	-	-	-	-	-
		5	-	-	-	-	-	-

PP - Number of Perishable Products, SL - Length of the Shelf-life, # - Instance Number, P - Production Oriented Time Windows, C - Customer Oriented Time Windows, S - Standard Time Windows, L - Loose Time Windows, TS - Triangular Setup Structure, NTS - Non Triangular Setup Structure

Table 5.2 – Detailed costs for all instances using the I-BS-VRPTW and I-LS-VRPTW models.

PP	SL	#	P-S-TS			P-L-TS			P-S-NTS			C-S-TS			C-L-TS			C-S-NTS		
			Setup	Vehicle	Travel	Setup	Vehicle	Travel	Setup	Vehicle	Travel	Setup	Vehicle	Travel	Setup	Vehicle	Travel	Setup	Vehicle	Travel
1	3	1	725	1000	189	550	1250	234	533	1000	189	1450/1375	1000	189	1150	750	165	1133/1000	1000	189
		2	1100	1000	322	1100	750	280	1067	1000	288	1700/1350	1250	322	1300	750	319/280	1267	1000	322/288
		3	1050	1000	345	775	1000	345	1000	1000	345	1175	1000	345	1050	1000	338	1067	1000	345
		4	700	1000	303	550/800	1000/750	303/238	467	1000	259	1275/900	1250	306	875/700	1000	303	867/667	1250	306
		5	1000	1250	360	1000/900	1000	305/328	800/667	1000	305	1200/1125	1250	360	1200/1000	750	265/288	800	750	288
5	3	1	900	750	353	700	750	363	667	750	353	925/900	1250	448	900	750	353	800	1000	402
		2	925	1000	344	775	1000	344	667	750	355	1225	1250	386	1025	1000	379	733/667	1250	386
		3	525	1000	311	375/525	1000/750	301	667	750	249	1050	1250	311	900/850	1000	301	800/667	1250	311
		4	600	750	406	450	750	406	267	750	406	750	1250	484	750	750	406	267	1250	484
		5	775	750	355	625	750	355	533/467	750	350/355	950	1250	425	950	750	355	533	1250	425
2	3	1	1375/1125	1250	306	950	1250	306	1467/1067	1250	306	1800/1550	1250	306	1375/1125	1250	306	1267/1200	1250	306
		2	1175	1250	221	1175	1250	221	1133	1250	221	1375	1250	221	1575	1250	221	1333	1250	221
		3	950	1250	280	950	1250	280	733	1250	280	1450/1550	1250	280	1100	1250	280	1267/800	1250	280
		4	1025	1250	281	1025	1250	281	1000/933	1250	281	1275	1250	281	1175	1250	281	1000	1250	281
		5	1500	1250	314	1250	1250	314	1333	1250	314	1825/1350	1250	314	1100	1250	314	1267/1200	1250	314
5	3	1	1250	1250	420	975	1000	384	867/667	1000	384/396	1100	1000	384	950/800	750	360	667	1000	396/378
		2	1150	750	269	1000	750	269	1000/933	750	269	1100	1250	395	1150	750	269	1200	1000	336
		3	950	750	386	825	750	386	867	750	386	1350	1250	450	1350	1000	408	867	1250	450
		4	1250	750	392	850	750	392	867	750	392	1300	1250	394	1300	750	392	1200/1067	1250	394
		5	950	750	376	700	750	376	533	750	376	1100	1250	412	925	1000	412	600/533	1000	412

PP - Number of Perishable Products, SL - Length of the Shelf-life, # - Instance Number, P - Production Oriented Time Windows, C - Customer Oriented Time Windows, S - Standard Time Windows, L - Loose Time Windows, TS - Triangular Setup Structure, NTS - Non-Triangular Setup Structure

5.3.3 Solution Examples

In this subsection, illustrative examples of instances in which the I-LS-VRPTW overcomes the I-BS-VRPTW are shown. In each example, two Gantt charts are used to represent graphically the solutions. The top chart represents the Gantt chart of the I-BS-VRPTW solution and the bottom illustrates the I-LS-VRPTW solution. Customers are arranged according to their time-windows boundaries and vertically at the Gantt chart, from customer 1 to 5. Products 1, 2 and 3 are denoted by light grey, dark grey and dotted bars, respectively. Setup operations are in black colour bars. The shelf-lives of perishable products are represented by thin white bars starting at the beginning of the production process. The time-windows boundaries are indicated by two vertical lines delimiting delivery operations. The travel time from the depot (or customer) to a customer is represented by 45 degree downward hatch box and the opposite operation, from customer to depot, by a 45 degree upward hatch bar. Moreover, the jobs that were split are pointed out by an upward or a downward arrow in the respective I-LS-VRPTW graphical solution.

In example 1 of Figure 5.2 lot sizing can improve the solution of I-BS-VRPTW by reducing setup operations (St-). In the I-BS-VRPTW solution the setup sequence is (1, 2, 1, 2, 3, 1, 3, 2). With the lot-sizing flexibility, it is possible to better use the shelf-life limitation of product 2 for customer 2 and rearrange the production sequence by sizing the lot of product 1 for customer 2. Thus, the new setup sequence is (1, 2, 1, 3, 2, 1), which entails two less setup operations, one for product 2 and one for product 3. The delivery operations are the same for both solutions.

Example 2 (Figure 5.3) is similar to example 1, but instead of reducing the number of setup operations, lot sizing has enabled a modification of the setup sequence (Seq), resulting in a lower solution cost. This example shows the importance that lot sizing can have when setup costs do not obey to the triangular inequality. It is noticeable that the changeover from product 1 to 2 is more economic if product 3 is produced in between. The lot-sizing operation allows for such setup sequence, while the products are still delivered without getting spoiled. Moreover, by sizing the lot of product 1 for customer 2 it was possible to reduce one setup for product 2.

In the example of Figure 5.4, the difference between batching and lot-sizing solutions is once again related to the reduction of setups. However, in this case, the delivery operations were also changed (Dist+, St-). The splitting of job (3,3) - product 3 for customer 3 - allowed a single batch production of product 2. This production change yields a different routing maintaining the same number of vehicles. Hence, the reduction of the setup costs counterweights the increase of the distance travelled.

Figure 5.5 shows an instance where the travel costs decrease due to the routing change provided by lot sizing (Dist-) and the setup costs remain unchanged. The batching solution uses a vehicle for supplying customers 1 and 4 and another for customers 2 and 3. When lot sizing is allowed, customers 1 and 3 are part of the same vehicle's route while customers

Figure 5.2 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=4, C-S-TS (St-).

Figure 5.3 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=4, C-S-NTS (Seq).

Figure 5.4 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=5, P-L-TS (Dist+, St-).

2 and 4 belong to other.

Figure 5.5 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=2, C-L-TS (Dist-).

Figure 5.6 illustrates the improvement of a batching solution by means of the reduction of one vehicle (V-, Dist-, St+). With the splitting of job (1,2) - product 1 for customer 2, it is possible to serve customers 1 and 4 along the same route. It is interesting to note

that in this solution, the usage of customers' time-windows up to the boundary. In the batching solution, only customers 3 and 4 share a vehicle's route, while all the others are supplied by different vehicles. On the other hand, the lot-sizing solution only uses three vehicles, also reducing the travel costs. However, more setup operations are needed increasing the total setup costs (that does not surpass the distribution costs decrease). In the batching solution, the setup sequence is $(1, 2, 3, 2)$, against $(1, 3, 2, 1, 3, 2)$ of the lot-sizing model.

Figure 5.6 – I-BS-VRPTW and I-LS-VRPTW solutions to instance PP=1, SL=3, #=4, P-L-TS (V-, Dist-, St+).

From the overall analysis of the detailed examples, it is clear now that the advantage of lot sizing is related to the better exploration of products' shelf-lives. Both instances with customer oriented time-windows (C) and non-triangular setups (NTS) increase the flexibility of the production process, therefore improving the efficacy of lot sizing.

5.4 Conclusions

In this chapter, we have analysed the importance of considering sizing the lots (or in other words, splitting the jobs) besides pure batching at the operational production and distribution planning when considering perishability. The lot-sizing decision is a counter-intuitive one in make-to-order environments and this is the first time that its importance is analysed. The logistic setting of our operational problem encompasses multiple perishable products subject to sequence dependent changeovers, which have to be delivered in a certain route by one of the available vehicles. We have developed models for integrating

with accuracy both lot sizing and batching with the vehicle routing problem with time-windows. In order to understand the impact of the extra flexibility coming from the possibility of splitting the lots, experiments varying different key parameters are designed and the solutions between the batching and lot-sizing models are compared.

Computational results for the set of systematically generated instances show that lot sizing is able to decrease the integrated production and distribution costs on very different types of instances. Both customer oriented time-windows and production environments with non-triangular setups seem to favour the importance of considering lot sizing in this operational problem. Several mechanisms to improve the batching solution were found by the lot-sizing model. The lot-sizing solution could achieve a better performance by: reducing the number of setups, changing the sequence, reducing setup costs, reducing the number of vehicles and/or the total travelled distance.

Future work should focus on strengthening the I-LS-VRPTW formulation and on developing efficient solution methods to solve this challenging and important problem.

6 ALNS for the operational integrated production and distribution problem of perishable products¹

Production and distribution problems with perishable goods are common in many industries. There, the finished perishable products should not take long to be delivered after production to satisfy the orders of their customers. Depending on the lifespan of the good, the production scheduling and distribution planning decisions should be taken jointly. Moreover, the competitiveness of the companies depend on the integration level of supply chain planning of products with restricted lifespan. Particularly at the operational level, the sizing and scheduling of production lots have to be decided together with vehicle routing decisions to satisfy the customers. However, such joint decisions make the problems hard to solve for industries with a large product portfolio.

From long-term to short-term decisions, the integrated production and distribution planning (*PDP*) is a common topic in the research literature. Many reviews categorize the papers of the topic, such as Vidal & Goetschalckx (1997), Sarmiento & Nagi (1999), Erengüç *et al.* (1999), Goetschalckx *et al.* (2002), Bilgen & Ozkarahan (2004), Chen (2004), Chen (2010), Schmid *et al.* (2013). The former surveys focused on the integration of such decisions at strategic and tactical levels, mainly on the design of the supply chain networks and inventory planning. However, the presence of perishable goods drives us to centre the discussion on the operational level, as short term decisions are crucial to the freshness/quality of the final products. Bilgen & Ozkarahan (2004) and Chen (2010) discuss the integrated *PDP* at the operational level. Both reviews argue that the publications in this area are recent and an emerging attention to these problems is necessary. The second review introduces some planning problems related to perishable goods. The authors highlight that most of reported applications that consider time-sensitive products have their customer orders satisfied as soon as the manufacturing has been finished, without a routing method or split-deliveries to save transportation costs. From a transportation point of view, Schmid *et al.* (2013) reviewed extensions of the vehicle routing problem inside supply chain frameworks, which include the integration of distribution and production decisions as lot-sizing and scheduling.

Chapter 5 addressed the operational integrated production and distribution planning problem (*OIPDP*). A review of the literature was discussed in Section 5.1. However, in

¹ The contents of this chapter are consonants with the paper “ALNS for the operational integrated production and distribution problem of perishable products”, referenced by (BELO-FILHO *et al.*, under review).

all the aforementioned papers, many features from the production environment are neglected, such as the cost and time consumption incurred by the setup operations and the sizing/splitting of the lots instead of the traditional batch production. In our case, a detailed production plan is crucial for an integrated *PDP*, due to the perishability of the products, which requires a proper calculation of the operation times to maintain freshness/quality. Therefore, two novel formulations were introduced considering such features and addressing two distinct structural assumptions regarding the lot size of production orders. The first model considers batching decisions for production orders, *i.e.*, the production orders are composed of full demand orders. The second formulation assumes lot-sizing/splitting decisions, in which a demand order may be manufactured in at least one production order. The latter approach proved to improve solutions of the former, being capable of reducing production and distribution costs. The last chapter did not propose a solution approach, as the focus was on showing that for industries with perishable products an integrated production and distribution planning is imperative and that lot-sizing/splitting flexibility should not be neglected. Nevertheless, the results indicate that the inherent complexity of the model does not allow the MILP-solver and the novel formulations to address real-world instances. The present chapter fulfils this gap by proposing solution methods suitable to tackle large-size instances that appear in practice.

As Schmid *et al.* (2013) emphasize, there is a lack of combined modelling and solution approaches (such as efficient metaheuristics) for integrated problems. According to Amorim *et al.* (2013a), even small instances may not be solvable to optimality using mixed integer linear programming solvers (MILP-solvers) in reasonable time. In this chapter, some methods to tackle large instances of this problem are proposed. An initial solution is generated by a speed-driven heuristic. From this solution, three distinct approaches are provided. The first uses a standard MILP-solver with the initial solution injected into the branch-and-bound tree. The second and the third methods are based on fixing some partitions of the solution and solving the remaining sub-problems. The second method is the fix-and-optimize (*FO*) method traditionally used for production planning problems. The third method is based on a large neighbourhood search (*LNS*) framework, proposed by Shaw (1998) and later improved by Ropke & Pisinger (2006), who introduced some adaptiveness to the *LNS* (*ALNS*). This method achieves successful results for transportation problems, as the case of the vehicle routing problem with time-windows. It has also been used in production planning problems providing good results (MULLER *et al.*, 2012). This approach destroys part of the solution and repairs it consecutively, in the hope of achieving new and improved solutions. The destroy and repair methods may be diverse and their combination can be chosen adaptively. Therefore, the most successful operators tend to be chosen more frequently, as different problems may need different strategies to yield a good solution.

The remaining of the chapter is organized as follows. Section 6.1 provides the definition

of the problem, together with a mathematical formulation. Section 6.2 details the proposed solution methods, such as constructive heuristic, exact methods, fix-and-optimize and *ALNS*. Section 6.3 compares the computational performance of the developed approaches for a set of generated instances. Finally, the conclusions and perspectives of research are presented in Section 6.4.

6.1 Problem statement

This section defines the operational integrated production and distribution problem (*OIPDP*), as seen in Chapter 5. The *OIPDP* consists of L parallel lines which produce P products (items) ordered by N customers. These customers must receive their product orders by a set of V vehicles. The products are manufactured on lines with limited capacity. The demand is deterministic and the ordered products incur production times and costs. Since equipment needs to be reconfigured for the production of different products, setup times and costs are assumed. The setups may be incurred in cleansing operations and when the environment changes (temperature, water level, tools), which determine their dependence on the sequence of products. At the beginning of the planning horizon, all the lines are set up for a product. The customer order may aggregate several products. The planning horizon of each line is split into time-varying production slots, in which both setup operations and production lots are accounted for.

The distribution is performed by capacitated vehicles that deliver products to multiple customers. A customer order has to be satisfied within a strict time-window with a single delivery. The fleet has at least the same number of vehicles as the number of customers, which guarantees an available vehicle for each customer. However, the use of a vehicle incurs a fixed cost. The travel times and costs are accounted and routing decisions should be made so that fewer vehicles are used and travel costs are minimized. The delivery operation starts by loaded vehicles in the depot. The vehicles then deliver the orders to the assigned customers within the customer time-windows. In each customer, a service time is considered. In the end the vehicles return to the depot.

The perishability of some products (P^* out of P) is determinant to the integration of the production and distribution processes. The shelf-life of perishable products is shorter than the planning horizon time. A customer's order must be met in perfect conditions, *i.e.*, the delivery should be within the lifespan of the manufactured products. The lifespan of a perishable product starts at the same time the production operation starts, after an occasional setup changeover operation.

6.1.1 Mathematical formulation

Here we present a MILP formulation for the *OIPDP* developed in (AMORIM *et al.*, 2013b). For timing decisions and constraints, the completion times of the production

operations were measured instead of their starting time. The parameters and decision variables are shown below.

Parameters

P (P^*)	Number of products (perishable)
L	Number of lines
N	Number of customers
V	Number of vehicles
S_l	number of slots for line l
dem_{jc}	demand for product j at customer c (units)
m_{lj}	minimum lot size for product j on line l
$cp_{lj}(tp_{lj})$	production cost (time) per unit of product j on line l
$scb_{lij}(stb_{lij})$	sequence-dependent setup cost (time) of a changeover from product i to product j on line l
α_l	initial product set up on line l
sl_j	shelf-life of product j (time)
Cap_l	available capacity (= latest completion time) of production line l
$CapV$	vehicle capacity on each trip
s_c	service time of customer c
$ct_{cd}(tt_{cd})$	cost (time) of travelling from customer c to d
$\bar{f}t$	fixed cost associated with each vehicle k
$[a_c, b_c]$	time-window for customer c

Decision Variables

q_{ljs}^c	quantity of product j produced in slot s on line l to serve customer c
y_{ljs}	equals 1, if line l is set up for product j in slot s (0 otherwise)
z_{lijs}	equals 1, if a changeover from product i to product j takes place at the beginning of slot s on line l (0 otherwise)
ct_{ls}	completion time of production slot s on line l
λ_{ljs}^c	equals 1, if there is production of product j for customer c in production slot s on line l (0 otherwise)
$ctls_c$	minimum completion time of the lifespan of the perishable products of customer c
$ctco_c$	completion time of the production of demand order of customer c
x_{cd}^k	equals 1, if arc (c, d) is used by vehicle k (0 otherwise)
w_c^k	starting time at which vertex c is serviced by vehicle k

The objective (6.1) is to minimise the sum of production and distribution costs. The production costs are composed of sequence-dependent setup and production costs. The distribution costs consist of fixed vehicle usage costs and distance-proportional costs. The model constraints are intentionally divided into three main groups: production, distribution and timing constraints.

$$\text{Min } \sum_{l,i,j,s} scb_{lij} z_{ljs} + \sum_{l,j,s,c} cpl_j q_{ljs}^c + \bar{f}t \sum_k (1 - x_{0,n+1}^k) + \sum_k \sum_{c,d} ct_{cd} x_{cd}^k \quad (6.1)$$

Production constraints

$$\text{s.t. } \sum_{l,s} q_{ljs}^c = dem_{jc}, \quad \forall j, c, \quad (6.2)$$

$$\sum_{i,j,s} stb_{lij} z_{ljs} + \sum_{j,s,c} tpl_j q_{ljs}^c \leq Cap_l \quad \forall l, \quad (6.3)$$

$$\sum_c q_{ljs}^c \geq m_{lj} (y_{ljs} - y_{lj,s-1}), \quad \forall l, j, s, \quad (6.4)$$

$$\sum_c q_{ljs}^c \leq \frac{Cap_l}{tpl_j} y_{ljs}, \quad \forall l, j, s, \quad (6.5)$$

$$q_{ljs}^c \leq dem_{jc} \lambda_{ljs}^c, \quad \forall l, j, s, c, \quad (6.6)$$

$$y_{l\alpha_l,0} = 1, \quad \forall l, \quad (6.7)$$

$$\sum_j y_{ljs} = 1, \quad \forall l, s, \quad (6.8)$$

$$z_{ljs} \geq y_{li,s-1} + y_{ljs} - 1, \quad \forall l, i, j, s \quad (6.9)$$

Constraints (6.2) set that the demand order of a customer for a product must be met by one or more production slots of different lines. However, the production is limited by the capacity constraints (6.3), which take sequence-dependent setup times into account. The production lot sizes are bounded by constraints (6.4)-(6.6). Constraints (6.4) set the minimum lot size for production in slots which require changeover setup. The maximum lot size is also limited by the capacity of the production line (6.5). The production lot size is bounded by the size of the customer order (6.6). Equations (6.7) and (6.8) determine the line configuration throughout the horizon. The changeovers are traced by constraints (6.9).

Distribution constraints

$$\sum_k \sum_d x_{cd}^k = 1, \quad \forall c, \quad (6.10)$$

$$\sum_d x_{0d}^k = 1, \quad \forall k, \quad (6.11)$$

$$\sum_c x_{cd}^k = \sum_c x_{dc}^k, \quad \forall k, d, \quad (6.12)$$

$$\sum_c x_{c,n+1}^k = 1, \quad \forall k, \quad (6.13)$$

$$\sum_{(j,c)} dem_{jc} \sum_d x_{cd}^k \leq CapV, \quad \forall k, c, \quad (6.14)$$

The distribution constraints account for vehicle assignment and routing. The depot assumes two indexes, 0 and $n+1$, the former is exclusive for vehicle departure operations and the latter index for vehicle arrival operations. Equations (6.10) assign a single vehicle per route. Constraints (6.11)-(6.13) set that the vehicle route should start at the depot (6.11), maintain its route after achieving a given node (6.12) and then return to the depot at the end of the trip (6.13). The load of each vehicle is constrained by the vehicle capacity (6.14).

Timing constraints

$$ct_{l1} \geq \sum_j stb_{l\alpha_{lj}} z_{l\alpha_{lj}1} + \sum_{j,c} t_{plj} q_{lj1}^c, \quad \forall l, \quad (6.15)$$

$$ct_{ls} \geq ct_{l,s-1} + \sum_{i,j} stb_{lij} z_{lij s} + \sum_{j,c} t_{plj} q_{ljs}^c, \quad \forall l, s > 1, \quad (6.16)$$

$$ctco_c \geq ct_{ls} - Cap_l(1 - \sum_j \lambda_{ljs}^c), \quad \forall l, s, c, \quad (6.17)$$

$$ctls_c \leq sl_j + ct_{ls} - \sum_d t_{plj} q_{ljs}^d + Cap_l(1 - \lambda_{ljs}^c), \quad \forall l, j, s, c, \quad (6.18)$$

$$w_0^k \geq ctco_c - \max_l \{Cap_l\}(1 - \sum_d x_{cd}^k), \quad \forall k, c, \quad (6.19)$$

$$w_d^k \geq w_c^k + s_c + tt_{cd} - \max_l \{Cap_l\}(1 - x_{cd}^k), \quad \forall k, c, d, \quad (6.20)$$

$$a_c \sum_d x_{cd}^k \leq w_c^k \leq b_c \sum_d x_{cd}^k, \quad \forall k, c, \quad (6.21)$$

$$\sum_k w_c^k \leq ctls_c, \quad \forall c, \quad (6.22)$$

$$q_{ljs}^c, z_{lij s}, ct_{ls}, ctco_c, ctls_c, w_c^k \geq 0, \quad (6.23)$$

$$y_{ljs}, \lambda_{ljs}^c, x_{cd}^k \in \{0, 1\}. \quad (6.24)$$

The production and distribution planning are coupled by the timing constraints, which schedule both types of operations. The completion time of a production slot depends on the setup changeover times and the processing times proportional to lot sizes (6.15) and

(6.16). The completion time of a customer order is given by the maximum completion time of all slots that produce for this customer (6.17). The lifespan of a perishable product is tracked in constraints (6.18). Then, the vehicle with this customer load should depart from the depot after its completion time (6.19). The arrival of a vehicle depends on the starting time of the preceding node, its service time and the travel times to reach the current node (6.20). The arrival time in a customer node should obey the time-windows requirement of this customer (6.21) and respect the lifespan of the perishable products present in this order (6.22). The variables domain are given by (6.23) and (6.24).

6.2 Proposed Methods

This section presents the developed methods and their parameter tuning procedures. The first method is a constructive heuristic of great value because it was designed to be fast and serve other methods. Then, some traditional methods are taken into account for comparison, including: 1) the trial to solve the problem exactly using MILP-solvers; 2) the previous method with an initial solution injected into the branch-and-bound tree (also known as warm start); 3) the traditional fix-and-optimize method, considering the customer time-windows sequence in the planning horizon; and 4) the proposed Adaptive Large Neighbourhood Search (*ALNS*).

6.2.1 Constructive heuristic

Constructive heuristics may achieve first solutions to complex problems by many strategies. In general, the focus of such procedures are on obtaining good-quality solutions in short amount of time. Secondary objectives may be simplicity, robustness and flexibility, *i.e.*, a procedure guided by a simple idea of solution that obtains feasible solutions to all sort of problem instances and that may be extended/simplified to problem extensions or variants. However, it is hard to achieve heuristics with these features all together, since there is a trade-off between solution quality and computational times in practice.

The constructive heuristic (*Heur*) was designed to obtain a first feasible solution in a short amount of time, allowing the improvement methods to generate better solutions. The first solution is mandatory for both fix-and-optimize and *ALNS* methods and the MILP-solver may not even find a first solution in reasonable time. An initial solution may be injected to the MILP-solver to be improved by its own methods. *Heur* states some simple rules to obtain a solution, as described in the following.

The first rule is the batching policy, *i.e.*, a customer order for a product is entirely produced in one slot, without permitting the lot splitting. Each slot should be occupied by just one order. As the number of vehicles and customers is the same ($N = V$), one vehicle is assigned to each customer delivery. Moreover, the delivery occurs at the end of

the customer time-window. The third rule is the “first to come, first to serve” rule, that is, the customer whom has to be delivered first has its order manufactured first. Finally, the customer orders are systematically assigned to the lines, in order to perform line-assignment decisions. In this assignment, all the positive demand orders ($dem_{jc} > 0$) are sequenced by customer and product order in set Π . Each order in set Π is then assigned to each line, in a linear cyclical manner, *i.e.*, the first order is assigned to the first line, the next order is assigned to the following line (in case there is no next line, the assignment turns back to the first line), and so on until all orders have been assigned to the lines. Algorithm 6.1 describes the procedure of the constructive heuristic.

Algorithm 6.1: Constructive heuristic.

```

Set a sequence of customers  $\Phi$  in the ascending order of the end of time-window  $b_c$ 
Define size of  $\Pi$ ,  $\pi = 0$ 
for  $c = \{\Phi_1, \dots, \Phi_N\}$  do
    for  $j = \{1, \dots, P\}$  do
        if  $dem_{jc} > 0$  then
             $\pi = \pi + 1$ 
             $\Pi_\pi = (j, c)$ 
        end
    end
end
Set  $l = 1$ ,  $s = 1$ 
for  $i = \{\Pi_1 = (j_1, c_1), \dots, \Pi_\pi = (j_\pi, c_\pi)\}$  do
    Set  $q_{l,j_i,s}^{c_i} = dem_{j_i,c_i}$ 
     $l = l + 1$ 
    if  $l > L$  then
         $l = 1$ 
         $s = s + 1$ 
    end
end
for  $c = \{1, \dots, N\}$  do
     $x_{0c}^c = x_{c,N+1}^c = 1$ 
     $w_c^c = b_c$ 
end
From the fixed variables, determine the remaining integer variables  $y$ ,  $z$ ,  $\lambda$ ,  $x$ 
Solve the remaining linear problem to find the timing decision variables

```

The following illustrative example shows a small instance of the OIPDP. It consists of two lines ($L = 2$, lines $L1$ and $L2$) manufacturing three products ($P = 3$, items $P1$, $P2$ and $P3$) for four customers ($N = 4$, customers $C1$, $C2$, $C3$ and $C4$). The maximum number of vehicles that can be assigned to deliver the goods is four ($V = 4$, respectively vehicles $V1$, $V2$, $V3$ and $V4$). Each line is composed of 6 slots and both lines are capacitated to 400 time units. The production cost/time is null/unitary ($cp_{lj} = 0$ and $tp_{lj} = 1$, respectively) and the minimum lot size is 5 units ($m_{lj} = 5$). The setup times are given by 10 time units ($stb_{lj} = 10$); otherwise they are zero. The setup costs are proportional to the setup times

by the relation $scb_{lij} = 25 \times stb_{lij}$. At the beginning of the planning horizon, the lines are set up for product $P1$. The demand and the shelf-life of the products are detailed in Table 6.1. The service time is null ($st_c = 0$). The capacity of the vehicles is limited to 250 units and a fixed vehicle cost of 250 cost units is charged in case the vehicle is used for delivery. The travel times are equal to the travel costs and are shown in Table 6.2 along with the time-windows.

Table 6.1 – Demand (dem_{jc}) and Shelf-life (sl_j).

dem_{jc}	$C1$	$C2$	$C3$	$C4$	sl_j
$P1$	0	50	50	50	200
$P2$	50	0	50	50	400
$P3$	50	50	0	50	150

Table 6.2 – Travel costs (ct_{cd}) and times (tt_{cd}) and time-windows (a_c, b_c).

$ct_{cd}(tt_{cd})$	$C1$	$C2$	$C3$	$C4$
Depot	20.0	40.0	20.0	20.0
$C1$		20.0	20.0	40.0
$C2$			34.6	60.0
$C3$				34.6
a_c	150.0	200.0	250.0	250.0
b_c	200.0	250.0	300.0	300.0

The *Heur* procedure runs as follows. First, the “first to come, first to serve” rule is applied, *i.e.*, a sequence of the customers is made according to the ascending order of b_c . Then, the orders are assigned to the lines. Let the product orders be represented by (j, c) . Order $(P2, C1)$ is assigned to line $L1$, order $(P3, C1)$ is assigned to line $L2$, $(P1, C2)$ to $L1$ and so on, until $(P3, C4)$ to $L1$. Decisions on the sequence of the customer orders assigned to the same line are made. In this case, the sequence of orders $(P1, C4)$ and $(P3, C4)$ may be swapped in line $L1$. The production plan is shown in Figure 6.1, which illustrates a Gantt chart showing the production and setup operations over the two lines ($L1$ and $L2$). The setups are represented by the dark gray bars. The production operations are white bars and the processing lots are given by the representation (j, c, q_{jls}^c) , *i.e.*, the product and the customer numbers followed by the lot size. The distribution plan is simple, as a distinct vehicle is assigned to each customer and the delivery time is fixed by the upper bound of the customer time-windows ($W_c^c = b_c$). The solution in Figure 6.1 values 2450.0 cost units.

6.2.2 Exact Methods

The exact methods are simply the MILP-solver procedures with and without a given initial solution. Although the method is exact, there is a limitation on the computational

Figure 6.1 – Production plan given by the heuristic (*Heur*).

time available for the method. Therefore, optimal solutions may not be achieved and proven. Notice that without an initial solution, the MILP-solver may not even find a feasible solution. The constructive heuristic *Heur* provides the initial solution for the MILP-solver, on the expectation of improvement by the procedures available in the MILP-solver software package and evolution on the branch-and-bound tree. Differently from the other methods presented, lower bounds based on linear relaxation are calculated and updated, providing a measure of the quality of the solution.

An optimal solution of the example of Section 6.2.1 is illustrated in Figures 6.2 and 6.3 (production and distribution plans, respectively). The former presents a Gantt chart which depicts the production and setup operations over time. The latter identifies the delivery routes taken by the vehicles. The boxed depot node D is where all the delivery operations start and circle nodes $C1$, $C2$, $C3$ and $C4$ denote the customers. The arrows represent the travel, along with the starting and completion times of the travels. When the vehicle returns to the depot, the arrow information contains the vehicle index. The solution objective value is 1654.6 cost units.

Figure 6.2 – Production plan of the optimal solution.

Figure 6.3 – Distribution plan of the optimal solution.

6.2.3 Fix-and-Optimize

The fix-and-optimize approach starts with an initial solution and proceeds by systematically fixing part of the solution and resolving the rest, in order to improve the solution in that specific neighbourhood. Many strategies can be found in the literature

to best determine the sequence in which decision variables should be fixed or freed. For instance, time-based neighbourhoods are commonly developed for lot-sizing problems. In these cases, the sequences of freed neighbourhoods are chosen according to the time in which the decisions affect the solution. The fix-and-optimize approach usually relies on the strategy of overlapping neighbourhoods, *i.e.*, intersectioned neighbourhoods with common free variables.

The fix-and-optimize procedures tested here start from the initial solution provided by the heuristic described in Section 6.2.1. Then, a sequence of customers based on the ascending order of their delivery time-windows is stated. All the binary variables related to a customer are either freed or fixed. Two parameters are necessary: the size of the free neighbourhood in each iteration and the size of the overlapping variables. We may denote the fix-and-optimize procedures by FO_{x-y} , where x is the size of the free neighbourhood and y is the number of overlapping customers. Based on this customer sequence, the procedure starts by fixing all customer-related variables except those of the x first customers. After solving this sub-problem, it fixes all customer-related variables except those of the next x customers, considering y overlapping customers. This procedure continues until the subproblem for the last set of customers has been solved. Algorithm 6.2 shows the pseudo-code of this procedure. Figure 6.4 illustrates the fix-and-optimize procedures FO_{1-0} and FO_{3-1} . In these examples eight customers are represented by the ellipses. The grey ones represent the customers with free variables, whereas the white customers are fixed. Each arrow separates two consecutive iterations. The arrow with the suspension points indicates that some analogous iterations are not shown. The FO_{x-y} procedure ends after the i th iteration, which is given by $\left\lceil \frac{N-y}{x-y} \right\rceil$.

Algorithm 6.2: Proposed fix-and-optimize heuristic (FO_{x-y}).

Input: feasible solution sol ;
Parameters: x and y (neighbourhood size and overlap, respectively);
 $seq \leftarrow$ sequence of the customers on increasing delivery time-windows order;
for $c = 1 \rightarrow N$ **do**
 for $c' = 1 \rightarrow c - 1$ **do**
 | Fix all variables λ and X from sol of customer $seq(c')$;
 end
 for $c' = c + x \rightarrow N$ **do**
 | Fix all variables λ and X from sol of customer $seq(c')$;
 end
 Solve the subproblem;
 $c \leftarrow c + x - y$;
end

Figure 6.4 – Differences between FO_1_0 and FO_3_2 .

6.2.4 *ALNS*

The large neighbourhood search (*LNS*) metaheuristic, proposed by Shaw (1998), aims to improve the solution through destroy and repair operators. Given an initial solution, the destroy operators undo part of it, removing some stated decisions. An implicit neighbourhood is created and a repair operator searches for new solutions there, inserting new decisions based on the maintained/fixed ones. The adaptive large neighbourhood search (*ALNS*) is defined by Ropke & Pisinger (2006) as “an *LNS* heuristic that uses several competing removal and insertion heuristics and chooses between using statistics gathered during the search”. Pisinger & Ropke (2010) summarize the improvement of the *LNS* approach and its extensions with the keyword adaptiveness. The *LNS* approach has successfully solved many problems, including scheduling and transportation applications, which clearly justifies our interest. The first work, Shaw (1998), implements the *LNS* heuristic for the vehicle routing problem with time-windows (*VRPTW*). Ropke & Pisinger (2006) made it adaptive for the pickup and delivery problem. Amorim *et al.* (2014) also used the *ALNS* framework for the *VRPTW*, considering perishable goods, heterogeneous fleet and multiple time-windows. Muller *et al.* (2012) presented a hybrid *ALNS* for the lot-sizing problem with setup times using an MILP-solver in the repair phase, instead of a “pure” heuristic.

Our *LNS* approach is adaptive and hybrid. The initial solution to the *ALNS* is given by the heuristic proposed in Section 6.2.1. Given a solution, a destroy and a repair operator are chosen adaptively. A set composed of multiple destroy operators $d \in \Omega$ is proposed. All destroy operators determine that some integer decision variables are fixed and the rest remain free, *i.e.*, the destroy operators analyse the neighbourhood generated by the free decisions. The binary variables chosen to be fixed are λ and X , respectively, the production and distribution assignment variables. The single repair procedure is given by the MILP-solver restricted to a limited time. As the number of free binary variables affects the efficiency of the repair heuristic, the destroy operators may have a changing parameter $\sigma_d, d \in \Omega$, bounded inferiorly and superiorly. The adaptive parameters control the number of free variables, hence the size of the neighbourhood browsed by the repair heuristic. The current values of these parameters may be increased or decreased according to the run of the repair heuristic. In case the sub-problem is solved

by the repair procedure before the limited time, or the final optimality gap is smaller than a given percentage, a larger neighbourhood may be explored and the parameter is increased. In case the final optimality gap is bigger than a given percentage, the parameter is decreased. Otherwise, σ_d remains unchanged. The increase or decrease in the optimality gap value may be different according to the destroy operator. Thus, the adaptive parameters for the operators allow the *ALNS* to be run in distinct environments, avoiding an initial tuning of the operator parameters.

The acceptance criterion, different from Ropke & Pisinger (2006), Kovacs *et al.* (2011) and Amorim *et al.* (2014) that resort to a simulated annealing framework, is simply the acceptance of the new solution in case it is better than the current. Furthermore, the best solution is always inserted as a warm start for the new search. The new solution is never worse than the current one and the search procedure becomes faster. A better new solution (*newsol*) means that the new solution objective function value $c(\text{newsol})$ is strictly lower than $c(\text{bestsol})$. Our solution framework requests a different strategy from the current literature regarding the weights and the scores of each destroy/repair operator. As the repair operator is unique, all the weights and scores are related to the destroy operators, hence to the combination destroy/repair operators. The probability ϕ_d of choosing a destroy operator d is proportional to the weight ρ_d and given by (6.25). At the beginning, all probabilities ϕ_d , $d \in \Omega$ are set to one.

$$\phi_d = \frac{\rho_d}{\sum_{d' \in \Omega} \rho_{d'}} \quad (6.25)$$

In each iteration of the *ALNS*, a destroy/repair operator is chosen by the roulette wheel selection. Scores ψ_d , $d \in \Omega$ are set to zero at the beginning of the run and after every weight update. The scores are updated after each iteration by summing up one unit to the score ($\psi_d \leftarrow \psi_d + 1$). As the acceptance criterion does not accept worse solutions, the score changes only in case a new solution is found. Weights ρ_d , $d \in \Omega$ are updated every it_{up} iterations by (6.26), according to scores ψ_d and a parameter α , which controls the influence of new and historical information.

$$\rho_d = (1 - \alpha) \times \rho_d + \alpha \times \psi_d \quad (6.26)$$

The pseudo-code of the proposed *ALNS* method is described in Algorithm 6.3.

All the operators represent a neighbourhood in which a new better solution may be found. In fact, they focus on the different aspects of the solution that may be improved. As different instances are taken into account, some of the neighbourhoods may be more effective in distinct situations, as well as in different instants of the search. The adaptive parameters reward the effectiveness and success of the operators. The destroy operators and their adaptive parameters are defined in the following.

Operator *Cst*: This first operator aims to improve joint production and distribution decisions on customers with near time-windows. A sequence of the customers is determined according to the ascending order of their time-windows. Then, all the decisions on

Algorithm 6.3: Proposed *ALNS*.

Input: feasible solution sol ;
 $bestsol \leftarrow sol$;
iteration $it \leftarrow 0$;
repeat
 Choose destroy method $d \in \Omega$ using probabilities ϕ_d ;
 Fix variables λ and X according to d and size parameter σ_d ;
 $newsol \leftarrow MILP(d(sol))$;
 if $c(newsol) < c(bestsol)$ **then**
 $bestsol \leftarrow newsol$;
 update ψ_d ;
 end
 update σ_d and it ;
 if iteration it is a multiple of it_{up} **then**
 update ρ_d , ϕ_d and ψ_d ;
 end
until time limit has been reached;

the chosen consecutive customers are revisited. The number of customers is adaptively chosen between 2 and N , with a (de)increment of 1.

Operators *Cst-P*, *Cst2-P* and *Cst3-P*: These operators are analogous to operator *Cst*, except that all the distribution decisions are now fixed to the values of the incumbent solution. The *Cst2-P* and *Cst3-P* operators also constrain the number of products allowed to be freed to 2 and 3 products, respectively. So, instead of freeing variables λ_{ljs}^c (*Cst-P*), only those related to a set of randomly chosen products are freed. When the number of products allowed to have any change is lower, more customers may be involved in the decision process. These operators aim to reschedule and allow a broader neighbourhood for lot-sizing/splitting decisions.

Operators *Slt1-P*, *SltL-P*, *SltT-P*: These operators focus on rescheduling the decisions made in different production slots. For all the operators the distribution processes are fixed. Operator *Slt1-P* randomly chooses some slots of the same line. The production processes present in these slots are freed. The decisions of all the other slots and lines are fixed. Operator *SltL-P* selects the same number of adjacent slots in each line. Differently stated, the last operator, *SltT-P*, sets a period of time and the decisions of all the respective slots across all the production lines are freed. Parameter σ for *Slt1-P* stands for the total number of slots chosen. For operator *SltL-P*, σ refers to the number of slots taken in each line. Finally, the changing parameter for *SltT-P* is the size of the time interval used to choose the slots.

Operator *Dst*: This last operator resolves the distribution sub-problem with all the production decisions fixed. All the vehicles-related variables are freed. This particular operator does not have any changing parameters, because its solution is quite fast.

Table 6.3 summarizes some information of the proposed destroy operators. The first column lists the operators. The second and third columns define the type of planning

decisions these operators are focused on: production (Prod) and/or distribution (Dist), respectively. The remaining columns state how the σ_d parameter changes: the lower bound (σ_{LB}) and the upper bound (σ_{UB}) of the parameter and its (de)increment (σ_{+-}). For instance, operator *Cst3-P* tackles only the production planning and σ_{Cst3-P} changes the number of customers freed in each iteration from 3 to N customers, with (de)increments of one unit.

Table 6.3 – Destroy operators of the *ALNS*.

Operators	Prod	Dist	Parameter	σ_{LB}	σ_{UB}	σ_{+-}
Cst	✓	✓	customer	2	N	1
Cst-P	✓		customer	2	N	1
Cst2-P	✓		customer	3	N	1
Cst3-P	✓		customer	3	N	1
Slt1-P	✓		slot	5	$\max(S_l)$	1
SltL-P	✓		slot	2	$\max(S_l)$	1
SltT-P	✓		time	100	$\max(capi)$	50
Dst		✓	-	-	-	-

6.3 Computational experiments

This section tests the developed methods with a set of instances. In the following, the generation of the test instances is described. Afterwards, the computational results are shown and the developed methods are compared.

6.3.1 Data Generation

The instance generator presented in Amorim *et al.* (2013b) is extended since, to the best of our knowledge, there are no other instances for the *OIPDP*. Twenty combinations of *number of lines* (L), *products* (P), *customers* (N) and *perishable products* (P^*) were generated. A compact nomenclature for the combinations is denoted by the short name $lL_pP_cN_ppP^*$, given the parameters listed above. These combinations are divided into four groups, according to the number of binary variables inherent to the MILP model, which indicates the size of the problem: very small, small, medium and large. Table 6.4 shows all the combinations and the approximate number of binary variables. For each combination, five instances were generated, totalizing 100 instances.

The rest of the parameters are drawn as follows. First, the lines are considered identical, so the parameters of a line are equal to the other lines. For all products and lines $tp_{lj} = 1$, $cp_{lj} = 0$, $m_{lj} = 5$ and $\alpha_l = 1$. The number of production slots of each line S_l is set to the first integer greater or equal than $\frac{P \times N}{L}$ in order to ensure that all the necessary setups and deliveries are performed. 75% of the demand dem_{jc} is generated

Table 6.4 – Different combinations and the approximate number of binary variables (in thousands).

Type of instances	Very Small	Small	Medium	Large
	<i>l01_p03_c05_pp01</i>	<i>l01_p05_c10_pp02</i>	<i>l01_p05_c15_pp02</i>	<i>l01_p10_c15_pp03</i>
	<i>l01_p03_c05_pp02</i>	<i>l01_p05_c10_pp03</i>	<i>l01_p05_c15_pp03</i>	<i>l01_p10_c15_pp05</i>
		<i>l02_p05_c10_pp02</i>	<i>l02_p05_c15_pp02</i>	<i>l02_p10_c15_pp03</i>
		<i>l02_p05_c10_pp03</i>	<i>l02_p05_c15_pp03</i>	<i>l02_p10_c15_pp05</i>
		<i>l04_p05_c10_pp02</i>	<i>l04_p05_c15_pp02</i>	<i>l04_p10_c15_pp03</i>
		<i>l04_p05_c10_pp03</i>	<i>l04_p05_c15_pp03</i>	<i>l04_p10_c15_pp05</i>
# of binary variables (1000's)	0.5	4.3	10.4	28.7

from the uniform distribution in the interval $U[40, 60]$ and the remaining 25% is set to zero. The setup times stb_{lij} are given by $U[6, 10]$ and the setup costs scb_{lij} are computed as $scb_{lij} = 25.0 \times stb_{lij}$. There is no setup time or cost for setups between products of the same type. The line capacity Cap_l is determined by: $Cap_l = \frac{\sum_{jc} dem_{jc} \times t_{plj}}{0.8L} + max_tt$. We estimate that the capacity utilization of the production is 80% and the capacity is complemented with max_tt as the maximum travel time. The shelf-life of the perishable products (sl_j) is given by $\min\{0.3 \times Cap_l; max_tt + 75 \times U[2, 3]\}$.

The travel times and costs are assumed to be the same. First, all customers are randomly positioned in a square of locations from (0,0) to (100,100). The depot is located at point (50,50). Then, the Euclidean distance is calculated between all pairs of customers and the depot. The service times s_c are negligible. The number of available vehicles is set to N and the cost of using each vehicle $\bar{f}t$ is set to 250. The capacity of the vehicle is computed by $CapV = \frac{3 \times \sum_{jc} dem_{jc}}{N}$.

The last parameters are the time-windows of each customer (parameters a_c and b_c), which are calculated according to each customer orders. To generate these time-windows, we propose the procedure described in Algorithm 6.4. The algorithm assigns each order of a customer to a line, represented by vector A , which accounts for the cumulative times of the assignments. The time-windows are then generated according to the maximum time found in A , the travel time and a small amount of time to guarantee the deliveries, and to perform potentially the delivery of multiple customers by a single vehicle. In the description of the algorithm, the maximum setup time is denoted by max_stb and the value of the average demand element by av_dem .

All the 100 instances are tested for feasibility purposes with the heuristic which generates the first solution. In case a solution is not found, then a new instance is generated until feasibility has been achieved. The instances are available at <<http://paginas.fe.up.pt/~pamorim/OIPDP.htm>>.

Algorithm 6.4: Pseudo-code to generate time-windows

```
Allocate a vector with size L and null values  $A[L] = [0..0]$ ;  
Define a cyclical iterator  $it$  for  $A$ ;  
for  $c = 1; c \leq N; c++$  do  
    for  $j = 1; j \leq P; j++$  do  
        if  $dem_{jc} > 0$  then  
            Sum  $dem_{jc} + max\_stb$  to  $A[it]$  ;  
             $it++$ ;  
        end  
    end  
    Define  $aux = U[2, 8] \times 0.1 \times av\_dem$ ;  
    Find the maximum time on  $A$  ( $max(A)$ );  
    Set  $a_c = max(A) + tt_{0c} + aux$ ;  
    Set  $b_c = a_c + 40$ ;  
end
```

6.3.2 Computational results

All computational experiments were performed on a workstation with two four-core Intel Xeon E5504 at 2.00 GHz with 24 GB RAM, running Linux. CPLEX version 12.4 from IBM was used as the MILP-solver. The data generator described in Section 6.3.1 was used to obtain the instance set. The maximum computational time for all methods was 3600 seconds. The MILP-solver was set to the maximum of 4 threads, opportunistic mode, across the methods.

Due to the strong NP-hardness of the *OIPDP* (AMORIM *et al.*, 2013b), it is not possible to find even integer solutions to the small instances with MILP-solvers. Therefore, we relied on the constructive heuristic (*Heur*) to obtain the first solutions. These solutions were used as a starting point for the *OIPDP*, *i.e.*, they were injected into the branch-and-bound tree of the MILP-solver.

Three parameters are set for the *ALNS*: 1) operator time; 2) parameter α , which defines a proportion for the historical and new weights; and 3) parameter it_{up} , which determines the frequency at which the weights are updated. The latter parameter was fixed to 20 iterations. Each operator has a limited time to destroy and repair the solution. As operator *Cst* performs a search in the integrated environment, it is given 5 more seconds than the other operators, which have the same time limit. To find out the better time limit, some tests were performed considering $\alpha = 0$, *i.e.*, the operators have always the same probability to be chosen. The operator times of 5, 10, 15 and 20 seconds were tested and the results are shown in Table 6.5. The columns denote the best, average and worst results for all instances, respectively, and the standard deviation (SD) of the quality of the solution, which measures the robustness of the method. Giving 5 seconds to each operator seems to deliver the best performance: the *ALNS* gets better as the number of iterations increases. We then tested the influence of parameter α , considering the 5 second limit for

each operator. Parameter α is important to establish the weights and probability of an operator being chosen. A higher α implies a strategy more focused on the intensification, whereas a smaller α aims to diversify the solution. Table 6.6 shows the respective results. The columns are analogous to those of Table 6.5. The results show that a balanced value for α yielded the best average performance. So, the experiments indicate that the best configuration for the *ALNS* is to consider 5 seconds time limit for operators, $\alpha = 0.2$ and $it_{up} = 20$.

Table 6.5 – Results for the *ALNS* with different operator time limits.

Time	Best	Average	Worst	SD
5	6703.78	6929.45	7204.38	2.88%
10	6736.97	6952.50	7196.72	2.61%
15	6763.32	6988.95	7268.27	2.88%
20	6787.65	7030.08	7337.92	3.07%

Table 6.6 – Results for the *ALNS* with different α values.

α	Best	Average	Worst	SD
0.0	6703.78	6929.45	7204.38	2.88%
0.2	6710.81	6920.88	7186.87	2.91%
0.4	6749.47	6974.33	7228.49	2.85%
0.6	6761.19	7029.31	7362.38	3.35%

For this *ALNS* configuration, Table 6.7 shows the relative frequency of the number of improvements obtained by each destroy/repair operators along each run of the algorithm. The results are clustered regarding the different classes of instances. The first row depicts the overall average performance. The instances are then split by size: very small, small, medium and larger, number of lines and number of perishable products: PP– and PP+, which aggregate the instances from each type that have less and more perishable products, respectively. Operators *Cst3-P* and *Cst2-P* are the most effective, being responsible for 26.6% and 17.8% of the improvements, respectively. The improvements obtained by these operators are more significant than the ones delivered by operator *Cst-P* (6.9%), which is also concerned about the neighbourhood of production decisions of different customers with fixed distribution, though the number of products taken into account by operator *Cst-P* is not constrained. It seems more interesting to tackle each partition with more customers together with a limited number of items. It is worth of mention that the performance of the operators are dependent on its sequence throughout the search. Table 6.7 also presents some characteristics of the operators according to the problem parameters. Operator *Cst*, which allows for joint decisions on the production and distribution planning, is negatively affected by the increase of the number of products, as the size of the neighbourhoods and the number of customer-related free variables increase.

The opposite occurs with *Cst2-P* and *Cst3-P* operators, whose effectiveness in achieving better solutions is improved, as the number of products is fixed. Notice that, for very small sized instances, the performance of operators Cst-P and Cst3-P is similar. The slot-strategy operators are more affected by the number of lines. Moreover, operator *Slt1-P* seems more suitable for problems with fewer lines, as it chooses only one line each iteration to re-optimize the production planning decisions and, therefore, line assignment decisions are not modified. The same arguments apply for the opposite behaviour of operator *SltT-P*. With more lines and the same number of products and customers, the customer time-windows tend to be closer, expanding the solution space of the distribution planning and, potentially improving the performance of the operator *Dst*. This operator is also affected by the number of products as a larger product portfolio incurs in more production decisions. Finally, the number of perishable products has a small influence on the performance of the operators (the performance difference was never larger than 2%). In fact, there was no need for perishable-driven operators, since the current operators handle the operations with this type of products.

Table 6.7 – Performance evaluation of the operators of the *ALNS*.

Class	Cst	Cst-P	Cst2-P	Cst3-P	Slt1-P	SltL-P	SltT-P	Dst
Average	15.1%	6.9%	17.8%	26.6%	6.8%	3.9%	13.7%	9.2%
Very Small	20.9%	5.4%	10.6%	39.1%	6.1%	0.0%	8.2%	9.7%
Small	16.2%	7.7%	17.8%	23.7%	5.7%	1.3%	16.0%	11.6%
Medium	16.7%	7.7%	16.6%	22.8%	4.3%	5.2%	13.7%	13.0%
Large	10.7%	5.8%	21.3%	29.1%	10.5%	6.4%	13.3%	3.0%
101	14.8%	5.8%	20.4%	31.7%	10.6%	2.6%	8.0%	6.1%
102	15.2%	8.7%	17.3%	24.6%	5.5%	4.1%	15.1%	9.6%
104	15.5%	6.6%	14.7%	21.8%	3.0%	5.3%	20.0%	13.1%
PP–	14.4%	6.4%	18.7%	27.0%	6.6%	4.0%	14.2%	8.7%
PP+	15.9%	7.4%	16.8%	26.2%	7.0%	3.7%	13.2%	9.8%

Six fix-and-optimize configurations were tested: *FO_1_0*, *FO_2_0*, *FO_2_1*, *FO_3_0*, *FO_3_1* and *FO_3_2* (see Section 6.2.3 for the difference between these methods). The configurations were compared using the solution performance gap, *i.e.*, the difference between the incumbent solution objective value of each method and the best solution achieved by the methods compared, divided by the same best solution. The best results were achieved by configurations *FO_3_1* and *FO_3_2*, with an average solution performance gap of 4.03% and 3.14%, respectively. The most successful fix-and-optimize methods include only the procedures with overlapping, which shows the success of this feature. Larger neighbourhoods (three costumers optimized per iteration instead of less costumers) lead to better solutions on average.

We then benchmark the performance of *ALNS* against the other solution methods. Table 6.8 provides the average solution performance gaps of the best methods and the

optimality gap of the best solution achieved and the lower bound provided by the MILP-solver. The rows are analogous to Table 6.7 and denote the parameters of the instances. The first columns report the upper-bound given by the construction heuristic (*Heur*), MILP-solver (*MILP*), fix-and-optimize approaches and *ALNS*. As the *ALNS* is not deterministic, which influences the algorithm performance, we have run the *ALNS* five times for each instance with different random seeds and report the best, average and worst results. The last column depicts the optimality gap of the best solution obtained across all methods. Table 6.8 indicates that the *ALNS* method yielded the best results, with a large difference over the other methods and the best run of *ALNS* achieved the best solutions for all instances. The heuristic provides poor-quality solutions and the *MILP* performance is strongly affected by the size of the problem, number of lines and perishable items. Fix-and-optimize methods with larger neighbourhoods obtained a competitive performance compared to *MILP*, with better results for medium to large instances. The average solution performance gap show that even the incumbent solution found in the worst run of the *ALNS* method is better than the solutions achieved by the other methods, for most of the cases. The average optimality gap presented is large even for very small instances.

Table 6.8 – Average solution performance gap and the best optimality gap achieved.

	<i>Heur</i>	<i>MILP</i>	<i>FO-2-1</i>	<i>FO-3-1</i>	<i>FO-3-2</i>	<i>ALNS</i>			Opt.
						Best	Avg.	Worst	Gap
Average	118.8%	59.6%	25.6%	18.8%	17.8%	0.0%	3.1%	7.3%	64.7%
Very small	57.9%	0.0%	6.5%	0.0%	0.1%	0.0%	0.0%	0.2%	17.1%
Small	116.0%	15.5%	28.7%	19.9%	16.8%	0.0%	3.1%	7.6%	63.9%
Medium	134.6%	60.5%	35.0%	23.4%	22.1%	0.0%	4.3%	9.8%	64.1%
Large	126.1%	122.7%	19.3%	19.3%	20.3%	0.0%	3.1%	7.0%	81.9%
l01	81.5%	34.6%	10.2%	7.6%	6.6%	0.0%	1.7%	3.5%	62.5%
l02	119.0%	65.9%	21.9%	15.2%	14.7%	0.0%	3.5%	7.8%	72.0%
l04	168.2%	86.7%	49.7%	37.3%	35.7%	0.0%	4.7%	12.0%	60.2%
PP−	114.5%	56.2%	25.2%	19.2%	16.2%	0.0%	3.1%	7.4%	64.8%
PP+	123.1%	63.1%	25.9%	18.4%	19.3%	0.0%	3.1%	7.3%	64.5%

In order to have an overall perspective of the benchmark, we compare the methods using performance profiles (DOLAN; MORÉ, 2002). In these performance profiles, we can choose procedures ($s \in \mathcal{S}$) to be evaluated by a set of instances ($p \in \mathcal{P}$) using a performance measure (t), which, in our case, is the objective function. For each instance p , the smallest objective function value obtained by the compared methods becomes the reference for the performance ratio $r_{p,s}$ given by (6.27). Now, $\rho_s(\tau) : \tau \geq 0$ is defined as the proportion of instances solved by method s with performance ratio lower than or equal to τ . The chart represents function $\rho_s(\tau)$, which is the (cumulative) distribution

function for the performance ratio (DOLAN; MORÉ, 2002). Therefore, the closer the curve to the top left corner, the better.

$$r_{p,s} = \frac{t_{p,s} - \min \{t_{p,s} : s \in \mathcal{S}\}}{\min \{t_{p,s} : s \in \mathcal{S}\}} \quad (6.27)$$

$$\rho_s(\tau) = \frac{|p \in \mathcal{P} : r_{p,s} \leq \tau|}{|\mathcal{P}|} \quad (6.28)$$

Figure 6.5 illustrates the performance profile comparing the heuristic proposed in Section 6.2.1 (*Heur*), the MILP-solver with the heuristic solution as the initial solution (Section 6.2.2), the fix-and-optimize variants (*FO_2_1*, *FO_3_1* and *FO_3_2* of Section 6.2.3) and the *ALNS* (Section 6.2.4)). The chart shows the performance of the methods regarding the solution value ($r_{p,s}$ is the value of the solution achieved by method s in instance p). In the chart, τ ranges from 0% to 100% ($\tau \in [0, 1]$), *i.e.*, all the solutions worse than at least 2 times the best solution found are neglected. To infer the performance of the *ALNS* we use the grey area that is limited by the best and worst cases of the five test runs and we also depict the average run. The *ALNS* showed a superior performance, as the best run always achieved the best solution to $\mathcal{P}$. Moreover, the worst runs of the *ALNS* beat the other methods. The *FO_3_1* and *FO_3_2* fix-and-optimize methods had similar performance, better than the *FO_2_1*. The *MILP* solution is far from the best solution achieved. Naturally, the worst performance belongs to the constructive heuristic method. However, the aim of the heuristic is to provide a fast solution, commonly a solution with a poor quality, given the assumptions taken by the heuristic. The *Heur* and *MILP* procedures have 37% and 73% of their solutions with $r_{p,s} \leq 100\%$. The other methods always found a solution within $\tau = 100\%$. The worst *ALNS* case has a solution with a relative difference of 35.1% from the best solution found. For example, the *MILP* has 45% of the solutions with a performance within the same value ($\tau = 35.1\%$).

Figure 6.6 shows a comparison of the methods concerning the improvement in the solution along the running time. The improvement of a solution is measured relative to the heuristic solution, which is the common warm start for all the compared methods. The *ALNS* solution value is given by the average. For each method the overall average improvement at the end of the search is shown. The chart clearly highlights the better performance of the *ALNS* method, mainly at the beginning of the run. Although the improvement in the *MILP* solution is slower, in the end it becomes faster, probably due to the MILP-solver strategy. The fix-and-optimize methods show a similar behaviour, however the *FO_2_1* shows a better performance in the beginning, as it deals with smaller neighbourhoods. Notice that the neighbourhoods of the *FO_2_1* are also considered in the *ALNS* by operator *Cst*. This comparison reinforces the statement that the use of different neighbourhoods may result in longer improvements.

Figure 6.7 details the benchmark by plotting the performance of the average solution value relative to the warm start solution in time for the four classes of instances: very

Figure 6.5 – Performance evaluation of the proposed methods.

Figure 6.6 – Performance of the average solution value relative to the warm start solution in time.

small, small, medium and large. For the very small class, the proposed methods show similar performance, achieving the final solution in a short period of time. For the remaining cases, there is an unquestionable dominance of the *ALNS* over the other procedures. Averagely, even the worst run of the *ALNS* outperformed the other approaches (see Table 6.8). The *ALNS* convergence gets slower as the size of the instances increase, however the difference of the *ALNS* solutions to the other method solutions is augmented. The fix-and-optimize methods have an intermediate performance and, for larger instances, have shown a stair pattern, mainly for *FO_3_1* and *FO_3_2*, which means that the subproblems are not solved to optimality. The performance of the MILP-solver deteriorates significantly with the size of the problem, number of production lines and perishable products. From all tested instances, the MILP-solver has only found two provably optimal solutions to instances of the very small size class. Regardless of the instance class, the optimality gap given by the relative difference of the integer solution and the lower bound of the branch-and-bound tree is large. Even the very small size class showed an optimality gap of approximately 17,1% for most of the solution methods. The number of products and customers increase this gap, contrarily to the number of lines and perishable products. These results were expected as the mathematical model is based on the well-known general lot-sizing and scheduling problem formulation (FLEISCHMANN; MEYER, 1997), which enables the incorporation of details at the cost of delivering a weaker lower bound.

Figure 6.7 – Performance of the average solution value relative to the warm start solution in time, for different instance sizes.

Table 6.9 shows the mean computational times of the compared methods in seconds. The proposed heuristic achieves the result in less than one second. In general, MILP-

solver and *ALNS* have computational times greater than 3600 seconds. The fix-and-optimize methods depend on the size of the neighbourhood of each iteration. Thus, very small instances are solved faster than the larger instances. The table also indicates that instances with more perishable products are solved faster on average. With less perishable products, the number of distinct schedule solutions are greater, unlike when there are more perishable products, which reduces the flexibility of the schedule solutions.

Table 6.9 – Average computational times of the best methods.

	<i>Heur</i>	<i>MILP</i>	<i>FO_2_1</i>	<i>FO_3_1</i>	<i>FO_3_2</i>	<i>ALNS</i>
Average	0.35	3547.91	1483.27	2011.09	2317.99	3600.03
Very small	0.10	3077.26	1.02	9.19	8.39	3600.00
Small	0.16	3600.06	457.51	1011.65	2055.43	3600.01
Medium	0.23	3600.11	915.38	2088.62	2077.35	3600.02
Large	0.73	3600.46	3571.00	3600.30	3591.04	3600.08
l01	0.29	3469.47	1615.20	2092.42	2397.72	3600.03
l02	0.37	3600.21	1554.41	2352.82	2902.50	3600.04
l04	0.39	3600.21	1236.22	1560.92	1627.17	3600.04
PP–	0.37	3553.35	1724.20	2083.02	2406.09	3600.03
PP+	0.32	3542.48	1242.33	1939.16	2229.88	3600.03

6.4 Conclusion

In this chapter, the operational integrated production and distribution planning problem (*OIPDP*) with perishable products is considered. In the production planning, the decision-maker tackles scheduling, line-assignment and lot-sizing/splitting decisions. The distribution planning consists of a vehicle-routing problem with time-windows. Besides, the perishability is a hard constraint to the problem, which reinforces the joint planning of the production and distribution operations. The *OIPDP* is a very difficult problem, and so far unsolvable using exact methods even for problems with a small number of products and customers. An adaptive large neighbourhood search fed with a simple constructive heuristic is proposed here. This solution method is compared to some traditional approaches, as the *MILP-solver* limited by time and a fix-and-optimize with overlap based on customer decisions.

The *ALNS* outperformed the traditional methods, proving that approaches with clever adaptive intensification and diversification procedures may lead to good solutions, even for hard problems. The *ALNS* improvement speed is faster than the other approaches, obtaining better results in short running times for the instances generated. A point of paramount importance on the development of the *ALNS* heuristic is that, in general, the bigger the number of iterations, the better the heuristic performs. Thus, small time local

searches (5 seconds) were chosen and the sizes of the explored neighbourhoods were chosen adaptively, which improved the solutions achieved and the robustness of the proposed method, regardless the instance size. The various operators play a main role as they promote a search in different neighbourhoods resulting in good robust solutions.

Extensions of the *OIPDP* may include split deliveries and multiple customer time-windows. The problem can also be extended to consider a multi-period planning horizon and facing some tactical decisions. Such extension approximates OIPDP to the well-known inventory routing and production routing problems. As the products are perishable, the integration of the production planning with the resource planning is highly recommendable, guaranteeing good raw materials and final products with a better quality and more possibilities for decreasing the distribution costs. So far, the *ALNS* technique has been very flexible and destroy/repair operators may be easily manipulated for extended/closer problems, which makes the approach a suitable proposal of solution method for these problems.

7 Conclusion

Supply chain planning is a challenge for many production systems due to the complex integration of industry activities over hierarchical levels of decisions (strategic, tactical and operational). Supply chain planning deals with procurement, production, distribution and sales decisions. Production planning has a crucial impact on supply chain planning and a realistic and efficient modelling of these activities promotes gains that enhance the position of the company regarding other market players.

In this thesis, lot-sizing problems have been addressed in distinct production planning contexts. Different extensions of the problem were discussed in light of the literature. The discussion has led to the development of novel formulations, solution approaches and the integration of production planning and other supply chain activities such as distribution planning. The assumption of realistic features and integrated decision-making improved the reliability of the mathematical formulations to practice and the quality of the solution plans achieved.

We first focus on the capacitated lot-sizing problem with backlogging, setup carryover and crossover (*CLSP-BL-SCC*). The setup carryover and crossover allow the setup state of the production line to be maintained over successive periods, even if a setup operation is running. Two novel formulations were proposed, based on the well-known capacitated lot-sizing problem model and the facility location lot size variable reformulation (KRARUP; BILDE, 1977). In one of the proposed formulations, the setup variable was indexed by the periods in which the setup operation may start and end, which resulted in a more compact formulation. Two experiments were performed with distinct focus. The former was related to the setup crossover and its features. The results showed that modelling setup crossover is important, mainly when setup times consume a large amount of the period capacity. The latter experiment aims at comparing the novel formulations to a literature model. The proposed formulations outperformed the literature model, especially the one with disaggregated setup variable. The setup crossover also provided more flexibility on the management of production idle time and on the priori decision of the period size.

We then address the capacitated lot-sizing problem with setup carryover and perishable products (*CLSP-PP*). The shelf-life of the products is measured in terms of large-bucket periods, which indicates a lifetime of medium size. Two models were proposed, with a difference regarding the modelling of the production lot size variable: the classical formulation and the facility location reformulation (KRARUP; BILDE, 1977). Due to the perishability feature, the former model must adopt a *first-in-first-out* policy at the inventory. The latter modelling provides a simpler approach to tackle the shelf-life and to track the quality of the products to the customer. An instance set was generated, based on literature well-known instances. The MILP-solver performed well for half-hour runs,

however, feasible solutions were not achieved for all instances and most of them were not proven optimal. The computational results indicate the need for experimenting other solution approaches, which provides good-quality feasible solutions in a short amount of time.

In order to overcome the hurdles of the models, new efficient methods were proposed. A lagrangean heuristic was developed to *CLSP-PP*. Capacity constraints and other item-coupling constraints were relaxed and the resulting problem is partitioned in N (number of items) subproblems solvable by dynamic programming procedures based on Wagner & Whitin (1958) algorithm. Subgradient optimization is applied to solve the lagrangean dual problem, updating the lagrangean multipliers. A heuristic based on Trigeiro *et al.* (1989) is used to tackle the problem, recalling that perishability and setup carryover are assumed. The proposed approach achieved feasible solutions for all instances, with competitive solution costs and computational times. The lower bound provided by the lagrangean heuristic is better than the lower bound obtained by the MILP-solver, which results in better optimality gaps of the proposed approach.

Perishability issues challenge the silos of the supply chain. Therefore, the operational integrated production and distribution problem (*OIPDP*) was then defined and explored. One of the focus was on proving that lot-sizing/splitting decisions may impact the production and distribution planning over the current practice of make-to-order environments, which consider just batching decisions. Two novel formulations were proposed, assuming batching decisions and lot-sizing decisions, respectively. An instance set was generated with the purpose of testing those models on a set of parameters such as: the number of perishable products, the length of the shelf-life, the setup time and cost structure, the tightness of the time windows and the orientation of the time windows. Computational results on small instances proved that lot-sizing/splitting decisions allow for better results by means of distinct mechanisms.

The inherent complexity of integrated production and distribution planning models inhibits the MILP-solver to address practical instances. This issue is overcome by means of a new solution approach for *OIPDP*, based on the adaptive large neighbourhood search framework (*ALNS*). An instance set composed of larger instances was generated based on the data of Chapter 5. A comparison of the performance of *ALNS* and traditional methods such as MILP-solver and fix-and-optimize had shown that *ALNS* was able to yield better solutions in the same amount of computational time. The different neighbourhoods provided by the *ALNS* operators promoted an efficient and robust solution method in terms of the improvement of the solution quality, even for short computational times.

Summing up, the main contribution of the thesis is the attempt to assume real-world aspects to lot-sizing problems. The setup crossover and the perishability are characteristics of paramount importance in the supply chain planning. Consistent mathematical

models and good solution methods addressing these issues provide considerable improvement on such planning. Three main lot-sizing problems were discussed in this thesis: *CLSP-BL-SCC*, *CLSP-PP* and *OIPDP*. The former problem considered the setup crossover and proved the beneficial impact of assuming such feature on production planning problems with substantial setup capacity utilization. The disaggregated modelling of the setup variable on *CLSP-BL-SCC*, as far as we know, is an innovation on lot-sizing problem formulation and proved itself of great value, performing better results than a competitive literature model. The latter problems discussed perishability issues according to the finished product shelf-life. *CLSP-PP* proved suitable for products with medium-term shelf-life (measured in periods). When the products are highly perishable, an integrated approach is justifiable and production planning decisions are jointly taken with distribution decisions. Moreover, lot-sizing decisions on such complex problem permitted better solutions regarding the current practice of batching decisions. Novel formulations were proposed for all problems and non-exact methods proposed for the lot-sizing problems with perishability.

7.1 Perspectives

Further research is clustered in five main topics. The first topic emphasises the wide variety of features that may be considered in the production planning problems. Then, more specific research directions are discussed for the proposed problems: (a) *CLSP-BL-SCC*; (b) *CLSP-PP*; and (c) *OIPDP*. Finally, perspectives on solution methods are pointed out.

Lot sizing has a broad range of features/extensions that might be included in the modelling of production planning problems. This thesis emphasized the modelling of problems with setup crossover and perishable products. More features/extensions were implicitly handled in at least one problem, such as backlogging, setup carryover, sequence-dependent setup times and costs and parallel lines. Future research on production planning should deal with more complex real-world aspects. To name a few, new production environments may be adopted (parallel lines, flowshop, jobshop and the flexible versions of the latter two), problems may be extended to multi-level production systems (serial, assembly and general product structures) and different sale contracts, including aspects as backlogging and lost sales. Regarding the setup operations, sequence-dependent setup times and costs may be modelled (holding or not the triangular inequality).

The setup crossover was discussed in the context of a big-bucket lot-sizing model. It would be interesting to use the same idea for small-bucket formulations, since short periods would imply more setup crossover operations, resulting in better use of resource capacity. The disaggregation of the setup variable was performed for item-independent setups, though it may be straightly extended to sequence-dependent setups, holding the

properties of Section 2.2.3.

Perishable products may require different policies on inventory management and assumptions on modelling of perishability. Here, the shelf-life of the products was considered fixed, though short and medium-to-large shelf-lives were discussed. However, production planning problems with perishable products may be affected by the decaying of the products, caused by physical deterioration or even by obsolescence. Then, the depreciation of the products might not be fixed, but varying in time (discretely or continuously), with dynamic value of the products depending on their quality. These aspects of product perishability leads to the pricing problem, where the sale price of the finished product may influence the production planning decisions.

On *OIPDP*, an integration of production and distribution planning was devised, however, other stages of the supply chain might be integrated to the problem. For instance, some raw materials are perishable too, indicating that the management of the replenishment should be considered in the model. Other aspects of the distribution planning would be devised as well, such as split deliveries and multiple customer time-windows. On supply chain management, the diffusion of recycling and rework practices is changing the destination of spoiled products, which may be repaired or used for other purposes, instead of being discarded. The reasons for reutilization of these products might be economical or legal and this is a growing challenge on the supply chain management of perishable products.

Various techniques of exact and non-exact solution approaches were proposed in this thesis. Third party mixed-integer linear programming solvers were used in exact approach contexts and matheuristically, *i.e.*, in solution procedures that combines heuristic and exact methods. For instance, fix-and-optimize and adaptive large neighbourhood search algorithms were developed using the concept of mixing heuristic and exact methods. The proposed lagrangean relaxation heuristic faces a lot-sizing problem using subgradient optimization and a feasibility heuristic based on Trigeiro *et al.* (1989) was implemented. All these approaches may be improved in efficiency and extended to related problems. For instance, the lagrangean heuristic should be enhanced with local search metaheuristic procedures. The field of solution approaches is vast and many techniques are available in the literature that may be used to face the discussed problems of this thesis.

Bibliography

- ALMADA-LOBO, B.; KLABJAN, D.; CARRAVILLA, M. A.; OLIVEIRA, J. F. Multiple machine continuous setup lot-sizing with sequence-dependent setups. *Computational Optimization and Applications*, v. 47, n. 3, p. 529–552, 2010.
- AMORIM, P.; BELO-FILHO, M.; TOLEDO, F. M. B.; ALMEDER, C.; ALMADA-LOBO, B. Lot sizing versus batching in the production and distribution planning of perishable goods. *International Journal of Production Economics*, Elsevier, v. 146, n. 1, p. 208–218, 2013.
- AMORIM, P.; MEYR, H.; ALMEDER, C.; ALMADA-LOBO, B. Managing perishability in production-distribution planning: a discussion and review. *Flexible Services and Manufacturing Journal*, Springer US, v. 25, n. 3, p. 389–413, 2013.
- AMORIM, P.; PARRAGH, S. N.; SPERANDIO, F.; ALMADA-LOBO, B. A rich vehicle routing problem dealing with perishable food: a case study. *TOP*, v. 22, n. 2, p. 489–508, 2014.
- AMORIM, P. S.; ANTUNES, C. H.; ALMADA-LOBO, B. Multi-Objective Lot-Sizing and Scheduling Dealing with Perishability Issues. *Industrial & Engineering Chemistry Research*, v. 50, n. 6, p. 3371–3381, 2011.
- AMORIM, P. S.; GÜNTHER, H.-O.; ALMADA-LOBO, B. Multi-objective integrated production and distribution planning of perishable products. *International Journal of Production Economics*, v. 138, n. 1, p. 89–101, 2012.
- ANWAR, M. F.; NAGI, R. Integrated lot-sizing and scheduling for just-in-time production of complex assemblies with finite set-ups. *International Journal of Production Research*, v. 35, n. 5, p. 1447–1470, 1997.
- ARMSTRONG, R.; GAO, S.; LEI, L. A zero-inventory production and distribution problem with a fixed customer sequence. *Annals of Operations Research*, v. 159, n. 1, p. 395–414, 2008.
- BELO-FILHO, M. A.; AMORIM, P.; ALMADA-LOBO, B. ALNS for the operational integrated production and distribution problem of perishable products. *International Journal of Production Research*, p. 1–28, under review.
- BELO-FILHO, M. A.; TOLEDO, F. M.; ALMADA-LOBO, B. Models for capacitated lot-sizing problem with backlogging, setup carryover and crossover. *Journal of the Operational Research Society*, v. 65, n. 11, p. 1735–1747, 2014.
- BILGEN, B.; OZKARAHAN, I. Strategic tactical and operational production-distribution models: a review. *International Journal of Technology Management*, v. 28, n. 2, p. 151–171, 2004.
- BLOCHER, J. D.; CHAND, S.; SENGUPTA, K. The Changeover Scheduling Problem with Time and Cost Considerations: Analytical Results and a Forward Algorithm. *Operations Research*, v. 47, n. 4, p. 559–569, 1999.

- BOUDIA, M.; LOULY, M.; PRINS, C. A reactive GRASP and path relinking for a combined production–distribution problem. *Computers & Operations Research*, v. 34, n. 11, p. 3402–3419, 2007.
- BOWMAN, E. H. Production Scheduling by the Transportation Method of Linear Programming. *Operations Research*, v. 4, n. 1, p. 100–103, 1956.
- BRAHIMI, N.; DAUZÈRE-PÉRÈS, S. A Lagrangian heuristic for capacitated single item lot sizing problems. *4or*, 2014. Disponível em: <<http://link.springer.com/10.1007/s10288-014-0266-3>>.
- BRAHIMI, N.; DAUZÈRE-PÉRÈS, S.; NAJID, N.; NORDLI, a. Single item lot sizing problems. *European Journal of Operational Research*, v. 168, n. 1, p. 1–16, 2006.
- BRAHIMI, N.; DAUZÈRE-PÉRÈS, S.; NAJID, N. M. Capacitated Multi-Item Lot-Sizing Problems with Time Windows. *Operations Research*, v. 54, n. 5, p. 951–967, 2006.
- BRISKORN, D. A note on capacitated lot sizing with setup carry over. *IIE Transactions*, v. 38, n. 11, p. 1045–1047, 2006.
- BUER, M. V.; WOODRUFF, D.; OLSON, R. Solving the medium newspaper production/distribution problem. *European Journal of Operational Research*, v. 115, p. 237–253, 1999.
- BUSCHKÜHL, L.; SAHLING, F.; HELBER, S.; TEMPELMEIER, H. Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. *OR Spectrum*, v. 32, n. 2, p. 231–261, 2010.
- CAI, X.; CHEN, J.; XIAO, Y.; XU, X. Product selection, machine time allocation, and scheduling decisions for manufacturing perishable products subject to a deadline. *Computers & Operations Research*, v. 35, n. 5, p. 1671–1683, 2008.
- CAMARGO, V. C.; TOLEDO, F. M. B.; ALMADA-LOBO, B. Three time-based scale formulations for the two-stage lot sizing and scheduling in process industries. *Journal of the Operational Research Society*, Nature Publishing Group, v. 63, n. 11, p. 1613–1630, 2012.
- CASERTA, M.; VOSS, S. A math-heuristic Dantzig-Wolfe algorithm for capacitated lot sizing. *Annals of Mathematics and Artificial Intelligence*, v. 69, n. 2, p. 207–224, 2013.
- CATTRYSSE, D.; SALOMON, M.; KUIK, R.; Van Wassenhove, L. N. A Dual Ascent and Column Generation Heuristic for the Discrete Lotsizing and Scheduling Problem with Setup Times. *Management Science*, v. 39, n. 4, p. 477–486, 1993.
- CHEN, H.-K.; HSUEH, C.-F.; CHANG, M.-S. Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research*, v. 36, n. 7, p. 2311–2319, 2009.
- CHEN, Z.-L. Integrated production and distribution operations: Taxonomy, models, and review. In: SIMCHI-LEVI, D.; WU, S.; SHEN, Z. (Ed.). *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*. [S.l.]: Kluwer Academic Publishers, 2004. cap. 17, p. 1–32.

- CHEN, Z.-L. Integrated Production and Outbound Distribution Scheduling: Review and Extensions. *Operations Research*, v. 58, n. 1, p. 130–148, 2010.
- CHEN, Z.-L.; VAIRAKTARAKIS, G. L. Integrated Scheduling of Production and Distribution Operations. *Management Science*, v. 51, n. 4, p. 614–628, 2005.
- CHENG, C.-h.; MADAN, M. S.; GUNASEKARAN, A.; YIP, K.-Y. An algorithm using Lagrangean relaxation and decomposition for solving a capacitated lot-sizing problem. *International Journal of Mathematics in Operational Research*, v. 2, n. 2, p. 205–232, 2010.
- CHENG, C.-h.; MADAN, M. S.; MOTWANI, J.; YIP, K.-Y. A Lagrangean relaxation approach for capacitated lot sizing problem with setup times. *International Journal of Operational Research*, v. 18, n. 3, p. 272–303, 2013.
- CHIANG, W.-C.; RUSSELL, R.; XU, X.; ZEPEDA, D. A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. *International Journal of Production Economics*, Elsevier, v. 121, n. 2, p. 752–767, 2009.
- CLARK, A.; ALMADA-LOBO, B.; ALMEDER, C. Lot sizing and scheduling: industrial extensions and research opportunities. *International Journal of Production Research*, v. 49, n. 9, p. 2457–2461, 2011.
- De Bodt, M. Lot sizing under dynamic demand conditions: A review. *Engineering Costs and Production Economics*, v. 8, n. 3, p. 165–187, 1984.
- DENIZEL, M.; ALTEKIN, F.; SÜRAL, H.; STADTLER, H. Equivalence of the lp relaxations of two strong formulations for the capacitated lot-sizing problem with setup times. *OR Spectrum*, Springer Berlin / Heidelberg, v. 30, n. 4, p. 773–785, 2008.
- DIABY, M.; BAHL, H.; KARWAN, M.; ZIONTS, S. Capacitated lot-sizing and scheduling by lagrangean relaxation. *European Journal of Operational Research*, v. 59, n. 3, p. 444–458, 1992.
- DIABY, M.; BAHL, H. C.; KARWAN, M. H.; ZIONTS, S. A Lagrangean Relaxation Approach for Very-Large-Scale Capacitated Lot-Sizing. *Management Science*, v. 38, n. 9, p. 1329–1340, 1992.
- DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming*, v. 91, n. 2, p. 201–213, 2002.
- DREXL, A.; HAASE, K. Proportional lotsizing and scheduling. *International Journal of Production Economics*, Elsevier, v. 40, n. 1, p. 73–87, 1995.
- DREXL, A.; KIMMS, A. Lot sizing and scheduling-survey and extensions. *European Journal of Operational Research*, Elsevier, v. 99, n. 2, p. 221–235, 1997.
- ENTRUP, M.; GÜNTHER, H.; Van Beek, P.; GRUNOW, M.; SEILER, T. Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research*, Taylor & Francis, v. 43, n. 23, p. 5071–5100, 2005.

- EPPEN, G. D.; MARTIN, R. K. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations research*, Institute for Operations Research and the Management Sciences, v. 35, n. 6, p. 832–848, 1987.
- ERENGÜÇ, S.; SIMPSON, N.; VAKHARIA, A. J. Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, v. 115, n. 2, p. 219–236, 1999.
- FIOROTTO, D. J.; ARAUJO, S. A. de. Reformulation and a Lagrangian heuristic for lot sizing problem on parallel machines. *Annals of Operations Research*, v. 217, n. 1, p. 213–231, 2014.
- FISHER, M. An applications oriented guide to Lagrangian relaxation. *Interfaces*, v. 15, n. 2, p. 10–21, 1985.
- FISHER, M. L. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, v. 50, n. 12 Supplement, p. 1861–1871, 2004.
- FLEISCHMANN, B.; MEYER, H. The general lotsizing and scheduling problem. *OR Spectrum*, Springer, v. 19, n. 1, p. 11–21, 1997.
- GARCIA, J.; LOZANO, S. Production and vehicle scheduling for ready-mix operations. *Computers & Industrial Engineering*, v. 46, n. 4, p. 803–816, 2004.
- GARCIA, J.; LOZANO, S. Production and delivery scheduling problem with time windows. *Computers & Industrial Engineering*, v. 48, n. 4, p. 733–742, 2005.
- GEISMAR, H. N.; LAPORTE, G.; LEI, L.; SRISKANDARAJAH, C. The Integrated Production and Transportation Scheduling Problem for a Product with a Short Lifespan. *INFORMS Journal on Computing*, v. 20, n. 1, p. 21–33, 2008.
- GEOFFRION, A. Lagrangean relaxation for integer programming. In: BALINSKI, M. (Ed.). *Approaches to Integer Programming*. [S.l.]: Springer Berlin Heidelberg, 1974, (Mathematical Programming Studies, v. 2). p. 82–114.
- GLOCK, C. H.; GROSSE, E. H.; RIES, J. M. The lot sizing problem: A tertiary study. *International Journal of Production Economics*, v. 155, p. 39–51, 2014.
- GOETSCHALCKX, M.; VIDAL, C. J.; DOGAN, K. Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms. *European Journal of Operational Research*, v. 143, n. 1, p. 1–18, 2002.
- GUIGNARD, M. Lagrangean relaxation. *Top*, v. 11, n. 2, p. 151–200, 2003.
- GÜNTHER, H.-O.; GRUNOW, M.; NEUHAUS, U. Realizing block planning concepts in make-and-pack production using MILP modelling and SAP APO ©. *International Journal of Production Research*, v. 44, n. 18-19, p. 3711–3726, 2006.
- HAASE, K.; KIMMS, A. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, Elsevier, v. 66, n. 2, p. 159–169, 2000.
- HELD, M.; WOLFE, P.; CROWDER, H. P. Validation of subgradient optimization. *Mathematical programming*, v. 6, n. 1, p. 62–88, 1974.

- HINDI, K. S.; FLESZAR, K.; CHARALAMBOUS, C. An effective heuristic for the CLSP with set-up times. *Journal of the Operational Research Society*, v. 54, n. 5, p. 490–498, 2003.
- JANS, R. Improved lower bounds for the capacitated lot sizing problem with setup times. *Operations Research Letters*, v. 32, n. 2, p. 185–195, 2004.
- JANS, R.; DEGRAEVE, Z. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*, v. 177, n. 3, p. 1855–1875, 2007.
- JANS, R.; DEGRAEVE, Z. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, v. 46, n. 6, p. 1619–1643, 2008.
- KACZMARCZYK, W. Modelling Multi-Period Set-up Times in the Proportional Lot-Sizing Problem. *Decision Making in Manufacturing and Services*, v. 3, n. 1, p. 15–35, 2009.
- KACZMARCZYK, W. Modelling Set-up Times Overlapping Two Periods in the Proportional Lot-Sizing Problem with Identical Parallel Machines. *Decision Making in Manufacturing and Services*, v. 7, n. 1-2, p. 43, 2013.
- KARAESMEN, I. Z.; SCHELLER-WOLF, A.; DENIZ, B. Managing perishable and aging inventories: Review and future research directions. In: KEMPF, K. G.; KESKINOCAK, P.; UZSOY, R. (Ed.). *Planning Production and Inventories in the Extended Enterprise*. [S.l.]: Springer US, 2011, (International Series in Operations Research & Management Science, v. 151). p. 393–436.
- KARIMI, B.; Fatemi Ghomi, S.; WILSON, J. The capacitated lot sizing problem: a review of models and algorithms. *Omega*, v. 31, n. 5, p. 365–378, 2003.
- KOPANOS, G. M.; PUIGJANER, L.; GEORGIADIS, M. C. Optimal Production Scheduling and Lot-Sizing in Dairy Plants: The Yogurt Production Line. *Industrial & Engineering Chemistry Research*, v. 49, n. 2, p. 701–718, 2010.
- KOPANOS, G. M.; PUIGJANER, L.; MARAVELIAS, C. T. Production Planning and Scheduling of Parallel Continuous Processes with Product Families. *Industrial & Engineering Chemistry Research*, v. 50, n. 3, p. 1369–1378, 2011.
- KOVACS, A. a.; PARRAGH, S. N.; DOERNER, K. F.; HARTL, R. F. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling*, v. 15, n. 5, p. 579–600, 2011.
- KRARUP, J.; BILDE, O. Plant location, set covering and economic lotsizing: An $O(mn)$ algorithm for structured problems. In: COLLATZ, L. (Ed.). *Numerische Methoden bei Optimierungsaufgaben, Band 3, Optimierung bei Graphentheoretischen und Ganzzahligen Problemen*. Basel: Birkhäuser, 1977. p. 155–180.
- LEMARÉCHAL, C. Computational Combinatorial Optimization. In: JÜNGER, M.; NADDEF, D. (Ed.). *Computational Combinatorial Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 2241). cap. Lagrangian Relaxation, p. 112–156.

- LEMARÉCHAL, C. The omnipresence of Lagrange. *Annals of Operations Research*, v. 153, n. 1, p. 9–27, 2007.
- LOW, C.; YEH, Y. The impact of lot splitting in a single machine scheduling problem with earliness - tardiness penalties. In: MITSUISHI, M.; UEDA, K.; KIMURA, F. (Ed.). *Manufacturing Systems and Technologies for the New Frontier*. [S.l.]: Springer London, 2008. p. 293–296.
- LOZANO, S.; LARRANETA, J.; ONIEVA, L. Primal-dual approach to the single level capacitated lot-sizing problem. *European Journal of Operational Research*, v. 51, p. 354–366, 1991.
- MANNE, A. Programming of economic lot sizes. *Management Science*, v. 4, n. 2, p. 115–135, 1958.
- MENDEZ, C.; HENNING, G.; CERDA, J. Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. *Computers & Chemical Engineering*, Elsevier, v. 24, n. 9-10, p. 2223–2245, 2000.
- MENEZES, A. A.; CLARK, A.; ALMADA-LOBO, B. Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. *Journal of Scheduling*, v. 14, n. 2, p. 209–219, 2010.
- MEYR, H. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, v. 139, n. 2, p. 277–292, 2002.
- MILLAR, H.; YANG, M. An application of Lagrangean decomposition to the capacitated multi-item lot sizing problem. *Computers & operations research*, v. 20, n. 4, p. 409–420, 1993.
- MILLAR, H.; YANG, M. Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. *International Journal of Production Economics*, v. 34, n. 1, p. 1–15, 1994.
- MILLARD, D. *Industrial inventory models as applied to the problem of inventorying whole blood*. Tese (Doutorado) — The Ohio State University, 1959.
- MOHAN, S.; GOPALAKRISHNAN, M.; MARATHE, R.; RAJAN, A. A note on modelling the capacitated lot-sizing problem with set-up carryover and set-up splitting. *International Journal of Production Research*, v. 50, n. 19, p. 5538–5543, 2012.
- MULLER, L. F.; SPOORENDONK, S.; PISINGER, D. A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. *European Journal of Operational Research*, Elsevier B.V., v. 218, n. 3, p. 614–623, 2012.
- NAHMIAS, S. Perishable inventory theory: a review. *Operations research*, v. 30, n. 4, p. 680–708, 1982.
- NIEUWENHUYSE, I. V.; VANDAELE, N. The impact of delivery lot splitting on delivery reliability in a two-stage supply chain. *International Journal of Production Economics*, v. 104, n. 2, p. 694–708, 2006.

- ÖZDAMAR, L.; BARBAROSOGLU, G. An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. *International Journal of Production Economics*, v. 68, n. 3, p. 319–331, 2000.
- PAHL, J.; VOSS, S. Discrete Lot-Sizing and Scheduling Including Deterioration and Perishability Constraints. In: DANGELMAIER, W.; BLECKEN, A.; DELIUS, R.; KLÖPFER, S. (Ed.). *Advanced Manufacturing and Sustainable Logistics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, (Lecture Notes in Business Information Processing, v. 46). p. 345–357.
- PAHL, J.; VOSS, S. Integrating deterioration and lifetime constraints in production and supply chain planning: A survey. *European Journal of Operational Research*, Elsevier B.V., v. 238, n. 3, p. 654–674, 2014.
- PAHL, J.; VOSS, S.; WOODRUFF, D. Discrete lot-sizing and scheduling with sequence-dependent setup times and costs including deterioration and perishability constraints. In: *Hawaii International Conference on System Sciences*. [s.n.], 2011. p. 1–10. Disponível em: <http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=5718564>.
- PARK, S.; MILLER, K. Random number generators: Good ones are hard to find. *Communications of the ACM*, v. 31, n. 10, p. 1192–1201, 1988.
- PARK, Y. B. An integrated approach for production and distribution planning in supply chain management. *International Journal of Production Research*, v. 43, n. 6, p. 1205–1224, 2005.
- PISINGER, D.; ROPKE, S. Large neighborhood search. In: GENDREAU, M.; POTVIN, J.-Y. (Ed.). *Handbook of Metaheuristics*. [S.l.]: Springer US, 2010, (International Series in Operations Research & Management Science, v. 146). p. 399–419.
- POCHET, Y.; WOLSEY, L. A. *Production Planning by Mixed Integer Programming*. [S.l.]: Springer New York, 2006. (Springer Series in Operations Research and Financial Engineering).
- POTTS, C. N.; WASSENHOVE, L. N. V. Integrating Scheduling with Batching and Lot-Sizing: A Review of Algorithms and Complexity. *The Journal of the Operational Research Society*, v. 43, n. 5, p. 395–406, 1992.
- QUADT, D.; KUHN, H. Capacitated lot-sizing with extensions: a review. *4OR*, v. 6, n. 1, p. 61–83, 2007.
- ROBINSON, E. P.; LAWRENCE, F. B. Coordinated Capacitated Lot-Sizing Problem with Dynamic Demand: A Lagrangian Heuristic. *Decision Sciences*, v. 35, n. 1, p. 25–53, 2004.
- ROPKE, S.; PISINGER, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, v. 40, n. 4, p. 455–472, 2006.
- SAHLING, F.; BUSCHKÜHL, L.; TEMPELMEIER, H.; HELBER, S. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, v. 36, n. 9, p. 2546–2553, 2009.

- SAMBASIVAN, M.; YAHYA, S. A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers & Operations Research*, v. 32, n. 3, p. 537–555, 2005.
- SANTOS, C.; MAGAZINE, M. Batching in single operation manufacturing systems. *Operations Research Letters*, v. 4, n. 3, p. 99–103, 1985.
- SARMIENTO, A. M.; NAGI, R. A review of integrated analysis of production-distribution systems. *IIE transactions*, v. 31, n. 11, p. 1061–1074, 1999.
- SAVELSBERGH, M. Local search in routing problems with time windows. *Annals of Operations Research*, Springer, v. 4, n. 1, p. 285–305, 1985.
- SCHMID, V.; DOERNER, K. F.; LAPORTE, G. Rich routing problems arising in supply chain management. *European Journal of Operational Research*, Elsevier B.V., v. 224, n. 3, p. 435–448, 2013.
- SHAPIRO, J. A survey of Lagrangian techniques for discrete optimization. *Annals of Discrete Mathematics*, v. 5, p. 113–138, 1979.
- SHAW, P. Using constraint programming and local search methods to solve vehicle routing problems. In: MAHER, M.; PUGET, J.-F. (Ed.). *Principles and Practice of Constraint Programming - CP98*. [S.l.]: Springer Berlin Heidelberg, 1998, (Lecture Notes in Computer Science, v. 1520). p. 417–431.
- SOX, C. R.; GAO, Y. The capacitated lot sizing problem with setup carry-over. *IIE Transactions*, v. 31, n. 2, p. 173–181, 1999.
- SUERIE, C. Modeling of period overlapping setup times. *European Journal of Operational Research*, v. 174, n. 2, p. 874–886, 2006.
- SUERIE, C.; STADTLER, H. The Capacitated Lot-Sizing Problem with Linked Lot Sizes. *Management Science*, v. 49, n. 8, p. 1039–1054, 2003.
- SUNG, C.; MARAVELIAS, C. T. A mixed-integer programming formulation for the general capacitated lot-sizing problem. *Computers & Chemical Engineering*, v. 32, n. 1-2, p. 244–259, 2008.
- SÜRAL, H.; DENIZEL, M.; Van Wassenhove, L. N. Lagrangean relaxation based heuristics for lot sizing with setup times. *European Journal of Operational Research*, v. 194, n. 1, p. 51–63, 2009.
- TEMPELMEIER, H.; DERSTROFF, M. A lagrangean-based heuristic for dynamic multilevel multiitem constrained lot-sizing with setup times. *Management science*, v. 42, n. 5, p. 739–757, 1996.
- THIZY, J.; VAN WASSENHOVE, L. Lagrangean relaxation for the multi-item capacitated lot-sizing problem: A heuristic implementation. *IIE transactions*, v. 17, n. 4, p. 308–313, 1985.
- TOLEDO, F. M. B.; ARMENTANO, V. A. A Lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines. *European Journal of Operational Research*, v. 175, n. 2, p. 1070–1083, 2006.

TRIGEIRO, W. W.; THOMAS, L. J.; MCCLAIN, J. O. Capacitated lot sizing with setup times. *Management science*, JSTOR, v. 35, n. 3, p. 353–366, 1989.

VIDAL, C.; GOETSCHALCKX, M. Strategic production-distribution models: A critical review with emphasis on global supply chain models. *European Journal of Operational Research*, v. 2217, n. 96, 1997.

VIERGUTZ, C. *Integrated Production and Distribution Scheduling*. Tese (Doutorado) — Universität Osnabrück, 2011.

WAGNER, B. J.; RAGATZ, G. L. The impact of lot splitting on due date performance. *Journal of Operations Management*, v. 12, n. 1, p. 13–25, 1994.

WAGNER, H.; WHITIN, T. Dynamic version of the economic lot size model. *Management science*, JSTOR, v. 5, n. 1, p. 89–96, 1958.

WU, T.; SHI, L. Mathematical models for capacitated multi-level production planning problems with linked lot sizes. *International Journal of Production Research*, v. 49, n. 20, p. 6227–6247, 2011.

ZHU, X.; WILHELM, W. Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE transactions*, Taylor & Francis, v. 38, n. 11, p. 987–1007, 2006.

A Dolan-Moré Chart

In this appendix, we present the chart technique of Dolan & Moré (2002), which provides a performance evaluation of different approaches for optimization problems. The technique plots the cumulative frequency curve of multiple approaches, under ranges of a performance measure. Mathematically saying, be a set of procedures ($p \in \mathcal{P}$), a set of test instances ($i \in \mathcal{I}$) and a performance measure (t), for instance, the objective function value or the optimality gap. Given that the procedures may yield distinct values for the performance measure, a normalization of the performance measure is required for a better comparison. The normalization may be made using performance ratios. So, let $t_{p,i}$ be the measure value provided by procedure p for instance i . A performance ratio of a instance i is defined by (A.1).

$$r_{p,i} = \frac{t_{p,i} - \min \{t_{p,i} : p \in \mathcal{P}\}}{\min \{t_{p,i} : p \in \mathcal{P}\}} \quad (\text{A.1})$$

For instance, being t the solution objective function value, (A.1) provides the gap between the incumbent solution and the best solution achieved by all methods. It is important to notice that sometimes the measure $t_{p,i}$ may not be found (neither solution was achieved by p for i , for example) and therefore $r_{p,i}$ should get a bounded value (the largest ratio, for instance). Now, given $|\mathcal{I}|$ the number of instances, the function f_p stated by (A.2) provides the percentage of the sum of all instances which have yielded performance ratios less or equal parameter τ . Function f_p is the (cumulative) distribution function for the performance ratio (DOLAN; MORÉ, 2002).

$$f_p(\tau) = \frac{1}{|\mathcal{I}|} |i \in \mathcal{I} : r_{p,i} \leq \tau| \quad (\text{A.2})$$

The Dolan-Moré Chart technique is very useful because it provides a fair comparison of methods, even though they return infeasible solutions for some instances. The cumulated accounting for the instances within ranges of performance allow the reader of the chart to perceive the behaviour of the methods. Furthermore, the relative difference between procedures are highlighted in this chart, helping the evaluation of the approaches. A short example is given below.

A.1 Example

A example of Dolan-Moré chart is fully detailed in this section. Be three methods, A, B and C, which solved 20 instances. Table A.1 details the absolute solution values on the second, third and fourth columns. The minimum solution value found by all methods is given by the fifth column of Table A.1 and the remaining columns provide a performance ratio as (A.1).

Figure A.1 show the curves plotted with (A.2). The chart illustrates the performance of the solution value of the approaches relative to the other approaches. Methods A, B and C are denoted by solid, dashed and dotted lines, respectively. On the left of the chart, it is noticeable the number of instances in which the solution method has found the best solution. Methods A, B and C obtained 10, 8 and 3 best solutions, respectively. Notice that equal or very close

solutions may occur. Around 40%, method B surpass A, *i.e.*, B obtained more solution values with a gap to the best solution achieved less or equal 40% than A does. The top of the chart is only reached if the method obtained solutions for every instance. Approach C did not obtain solution for 10% of the instances and hence did not reach the top. Methods A and B clearly outperformed C, with better solutions for all ranges of gaps over the best solution achieved. Although for some instances C yielded better solution than both A and B, the important here is counting the cumulated performance of the instances to obtain a holistic performance of the method.

Table A.1 – Absolute and normalized solution value of three approaches.

Instance	A	B	C	Minimum	A	B	C
I1	14,3	20,2	—	14,3	0%	41%	—
I2	16,7	27,1	—	16,7	0%	62%	—
I3	85,2	44,9	84,4	44,9	89%	0%	87%
I4	56,3	29,5	78,1	29,5	90%	0%	165%
I5	56,0	32,9	44,3	32,9	70%	0%	34%
I6	79,6	61,9	50,3	50,3	58%	22%	0%
I7	62,1	35,7	76,6	35,7	73%	0%	114%
I8	9,4	13,4	20,4	9,4	0%	42%	117%
I9	17,0	29,7	20,8	17,0	0%	74%	22%
I10	54,9	83,5	35,0	35,0	57%	138%	0%
I11	26,1	44,8	72,6	26,1	0%	71%	178%
I12	73,4	62,2	96,2	62,2	18%	0%	54%
I13	59,8	37,3	67,4	37,3	60%	0%	80%
I14	33,6	15,9	36,9	15,9	111%	0%	132%
I15	56,8	56,8	79,3	56,8	0%	0%	39%
I16	52,0	70,7	45,5	45,5	14%	55%	0%
I17	70,6	97,1	79,6	70,6	0%	37%	12%
I18	88,5	36,7	80,5	36,7	141%	0%	119%
I19	61,4	86,3	79,2	61,4	0%	40%	28%
I20	35,0	69,2	75,0	35,0	0%	97%	114%

Figure A.1 – Performance chart for normalized solution values.