CAMPAIGN PLANNING UNDER SEQUENCE DEPENDENT FAMILY SETUPS AND CO-PRODUCTION IN PROCESS INDUSTRY

by

Serkan Kalay

B.S., Industrial Engineering, Galatasaray University, 2011

M.S., Industrial Engineering, Boğaziçi University, 2014

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Industrial Engineering

Boğaziçi University

2020

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to my supervisor, Dr. Z. Caner Taşkın, for his guidance always motivating and enabling the next step forward with his expertise throughout the writing of this dissertation. He has provided a great deal of support, patience and invaluable mentorship. Working following his lead has been an invaluable privilege to me.

I would like to thank ICRON Technologies and Şişecam for introducing me to the passionate field of glass production, from which, quoting a field expert, *"one cannot get away after seeing inside of the furnace."* I would also like to acknowledge valuable suggestions from Dr. Ümit Bilge and Dr. Dilek Tüzün Aksu during their participation in my dissertation committee.

I am grateful to my parents, my sister and my brother-in-law for their love, caring and being there for me no matter what I was chasing during my entire life. Finally, I am indebted to my deary wife, Esra, for her endless patience and support. She provided me all the comfort I needed to carry out this work. Together with my daughter, the apple of my eye, Ipek, they have been the happiest distraction I could ever ask for. Successfully completing this dissertation, I am looking forward to enjoy the rest of my life together with them.

ABSTRACT

CAMPAIGN PLANNING UNDER SEQUENCE DEPENDENT FAMILY SETUPS AND CO-PRODUCTION IN PROCESS INDUSTRY

We investigate tactical level production planning problem in process industries, with float glass manufacturing being the specific application domain. Process industries are cost intensive, and as a result, efficient usage of capacity through planning is necessary.

In the presence of high sequence dependent family setup costs, the need for planning production in batches, or campaigns as named in the float glass industry, arises. Campaign planning is determining timing and duration of each product family, which translates into setups. Moreover, availability of input data in different resolution, i.e. setup times in continuous time whereas customer demand forecast are available in discrete time, increases the complexity. Co-production is a phenomenon that exists in several industries including float glass manufacturing. Usually due to some special characteristic of the manufacturing process some products need to be produced by necessity. This is another challenge for efficient capacity usage as well as inventory management.

We study the problem for different complexity levels. We start with single machine instance and develop two formulations. A novel branch-and-price algorithm is proposed for the parallel machine extension. Finally, we extend the problem to multiple product hierarchy levels and network structure including customer locations. We demonstrate the efficiency of our methods through extensive numerical experiments as well as some further tests to analyze the sensitivity of the cost components.

ÖZET

PROSES SANAYİSİNDE SIRA BAĞIMLI ÜRÜN AİLESİ SETUPI VE BİRLİKTE ÜRETİM KOŞULLARINDA KAMPANYA PLANLAMA

Bu tezde proses sanayisinde taktik seviye üretim planlama problemini incelenmiş ve uygulamalar için bir proses sanayisi olan düz cam üretimi temel alınmıştır. Proses sanayisi maliyet odaklı olması sebebiyle kapasitenin planlama marifetiyle verimli bir şekilde kullanılması önem teşkil etmektedir.

Yüksek maliyetli sıra bağımlı ürün ailesi setuplarının varlığında düz cam sanayisinde kampanya olarak adlandırılan partiler halinde üretim ihtiyacı ortaya çıkar. Kampanya planlama bu partilerin zamanlamasını ve uzunluğunun belirlenmesidir ve eş zamanlı olarak da setup planları da oluşturulmuş olur. Setup sürelerinin sürekli zamanda müşteri taleplerinin ise kesikli zamanda ifade edilmesi problemin karmaşıklığını artırmaktadır. Birlikte üretim düz cam üretimi de dahil olmak üzere bazı sektörlerde bulunan bir olgudur. Genellikle imalat sürecinin birtakım özellikleri sebebiyle bazı ürünler zorunlu olarak üretilir. Bu durum kapasitenin verimli kullanılması için olduğu kadar stok yönetimi acısından da zorlayıcı bir başka etmendir.

Tez kapsamında problemin farklı zorluk dereceleri çalışılmıştır. İlk olarak tek üretim hattı incelenmiş ve iki adet matematiksel model geliştirilmiştir. Paralel hatların bulunduğu versiyon için yeni bir dal-fiyat algoritması geliştirilmiştir. Son olarak, problem çoklu ürün hiyerarşisi ve müşteri konumlarını da içeren şebeke yapısında çalışılmıştır. Sayısal deneyler önerilen yöntemlerin başarısını göstermiştir. Maliyet kalemlerine duyarlık deneyleri de yapılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A_t	Available capacity of the machine in period t in days
b_j	Cost of backlogging a demand of product j for a single period
b_{jl}	Cost of backlogging a demand of product j for a single period in facility l
B_t	Setup time, in days, spent at the end of period t
B_{pt}	Setup time, in days, spent for pattern p at the end of period t
c_{fg}	Setup cost of switching from family f to family g
c_{fgr}	Setup cost of switching from family f to family g on production line r
c_p	Total setup cost of family order within pattern p
c_{pr}	Total setup cost of family order within pattern p on production line r
d^o_{ft}	Number of days spent for production of family f in order o in period t
d^o_{frt}	Number of days spent for production of family f in order o on production line r in period t
D_{jt}	Demand of product j in period t
D_{jlt}	Demand of product j in facility l in period t
f^H_p	First family in pattern p
f^T_p	Last family in pattern p
$F(p)$	Set of families belonging to pattern p
F_t	Setup time, in days, spent at the beginning of period t
F_{pt}	Setup time, in days, spent for pattern p at the beginning of period t
F_{rt}	Setup time, in days, spent on production line r at the beginning of period t

F_{prt}	Setup time, in days, spent on production line r for pattern p at the beginning of period t
$F^o(p)$	Set of families appearing in order o in pattern p
h_j	Inventory holding cost for product j
h_{jl}	Inventory holding cost for product j in facility l
$I_{j(-1)}$	Beginning inventory of product j
$I_{jl(-1)}$	Beginning inventory of product j in facility l
I_{jt}	Inventory of product j at the end of period t
I_{jlt}	Inventory of product j in facility l at the end of period t
$J(f)$	Set of products belonging to family f
MD_f	Minimum production duration for family f in days
MD_{fr}	Minimum production duration for family f in days on production line r
MD_{fp}	Minimum production duration for family f in middle order of pattern p
MST_{fg}	Maximum setup time in days needed for switching from product family f to any other family or from any family to family g
MST_{fgr}	Maximum setup time in days needed for switching from product family f to any other family or from any family to family g on production line r
n^P_{fgt}	Number of days spent for setup in predecessor period $t-1$ for switched from family f to family g at the beginning of period t
n^P_{fgrt}	Number of days spent for setup in predecessor period $t-1$ for switched from family f to family g at the beginning of period t on production line r
n^S_{fgt}	Number of days spent for setup in successor period t for switched from family f to family g at the beginning of period t

n^S_{fgrt}	Number of days spent for setup in successor period t for switched from family f to family g at the beginning of period t on production line r
n_{jlm}	Unit transportation cost of product j from facility l to facility m
N^I	Number of integer periods
N^F	Number of fixed periods
NT_{fp}	Number of times family f appears in the middle order of pattern p
$P(f)$	Set of patterns containing family f at least once
$P^S(f)$	Subset of patterns whose first family is f
$P^E(f)$	Subset of patterns whose last family is f
$P^o(f)$	Set of patterns containing family f in order o
$Q(j)$	Index of the quality group of product j
R_{fqs}	Maximum production ratio/percentage for quality group q and size group s for family f
R_{fqsr}	Maximum production ratio/percentage for quality group q and size group s for family f on production line r
$S(j)$	Index of the size group of product j
S_{jtk}	Satisfied quantity of demand from period t of product j in period k
S_{jltk}	Satisfied quantity of demand in facility l from period t of product j in period k
ST_{fg}	Setup time needed for switching from product family f to family g in days
ST_{fgr}	Setup time needed for switching from product family f to family g on production line r in days
ST_p	Setup time needed for family order within pattern p in days
ST_{pr}	Setup time needed for family order within pattern p in days on production line r

STP_f	Maximum setup time in days such that product family f is predecessor
STP_{fr}	Maximum setup time in days such that product family f is predecessor on production line r
STS_f	Maximum setup time in days such that product family f is successor
STS_{fr}	Maximum setup time in days such that product family f is successor on production line r
T_{jlmt}	Transported quantity of product j from facility l to facility m in period t
u_{jr}	Unit production cost for producing product j on line r
U_{jt}	Unsatisfied quantity of demand from period t of product j
U_{jlt}	Unsatisfied quantity of demand in facility l from period t of product j
v_j	Production speed of product j, machine-days required for unit production
v_{jr}	Production speed of product j, machine-days required for unit production on production line r
X_{jt}	Production quantity of product j in period t
X_{jrt}	Production quantity of product j on production line r in period t
δ_{pt}	Binary indicator variable for selection of pattern p in period t
δ_{prt}	Binary indicator variable for selection of pattern p on production line r in period t
γ^S_{ft}	Indicator for selection of family f as starting in period t
γ^S_{frt}	Indicator for selection of family f as starting on production line r in period t
γ^E_{ft}	Indicator for selection of family f as ending in period t

γ^E_{frt}	Indicator for selection of family f as ending on production line r in period t
$\Gamma(f,g)$	Set of product family couples that are infeasible, $f,g \in F$
$\Gamma^r(f,g)$	Set of product family couples that are infeasible on production line r, $f,g \in F$
π_r	Dual variables associated with Equations (5.31)
μ_{frt}	Dual variables associated with Equations (5.32)
θ_{fgt}	Auxiliary variable indicating whether machine switched from family f to family g at the beginning of period t
θ_{prt}	Auxiliary variable indicating whether machine switched from pattern p to pattern r at the beginning of period t

LIST OF ACRONYMS/ABBREVIATIONS

BBS	Best Bound Search
BFS	Breadth-First Search
B&B	Branch-and-Bound
B&P	Branch-and-Price
CG	Column Generation
CP	Constraint Programming
CPM-EP	Extended Pattern Based Campaign Planning Model
CPM-EPMLN	Extended Pattern Based Campaign Planning Model on Multiple Level Network
DAG	Directed Acyclic Graph
DFS	Depth-First Search
DH	Descent Heuristic
DNS	Diminishing Neighborhood Search
DP	Dynamic Programming
DPM	Demand Projection Model
DDPM	Dynamic Demand Projection Model
DDPM-RNF	Dynamic Demand Projection Model with Relax-and-Fix
EDD	Earliest Due Date
FTB	Family Transition Based
FTBM	Family Transition Based Model
FTBMV	Family Transition Based Model Variant
FTBMV-PM	Family Transition Based Model Variant on Parallel Machines
FTBMV-MLN	Family Transition Based Model Variant on Multiple Level Network
GA	Genetic Algorithm
GLSP	General Lot Sizing Problem
IDP	Incomplete Dynamic Programming
LP	Linear Program

MTO	Make-to-Order
MIP	Mixed Integer Programs
MILP	Mixed Integer Linear Programs
MINLP	Mixed Integer Non-Linear Programs
PTB	Pattern Transition Based
PTBM	Pattern Transition Based Model
PTBMV	Pattern Transition Based Model Variant
PTBMV-PM	Pattern Transition Based Model Variant on Parallel Machines
RCPM-EPMLN	Restricted Extended Pattern Based Campaign Planning Model on Multiple Level Network
RCPM-EP	Restricted Extended Pattern Based Campaign Planning Model
R&F	Relax-and-Fix
RH	Rolling Horizon
RVNS	Reduced Variable Neighborhood Search
SA	Simulated Annealing
TS	Tabu Search
VNS	Variable Neighborhood Search

1. INTRODUCTION

Production planning is a decision process to determine how available resources including materials and machines should be allocated such that demand from customers is satisfied while conforming to the desired level of performance indicators as much as possible. Decisions include manufacturing time and quantity of each product in addition to the selection of machine or alternative that the manufacturing will be executed on.

Supply chain planning often directly considers manufacturing, transportation, inventory holding and demand satisfaction related costs. Nevertheless, loss of efficiency in production line capacity usage can have significant impact on the overall effectiveness, especially in process industries. For instance, furnaces used in float glass manufacturing need to be up and running 24/7 due to the continuous production nature of the process even if there is insufficient demand or stock targets. Moreover, the furnace must also keep running during setup, which can take several days. Since lost capacity is highly undesirable, an elaborated setup decision within the plan cycle is necessary.

In the presence of high associated costs, the duration of a production run for a given setup needs to be long enough so that the production plan ensures the balance between setup and inventory holding costs for products involved. Therefore, products belonging to a certain family are usually produced together in *campaigns*. In glass manufacturing for instance, products that have the same color, which is the main driver of setup, are produced in campaigns. For a specific color, the plan usually contains one or two campaigns in a year, in order to minimize the changeovers [2]. Hence, we can define campaign planning as the process of determining the timing and the length of such production run decisions, which also means determining setups.

We can define setup as time, cost and possibly material necessary to spend to start producing a scheduling unit, and we can categorize it in different aspects.

Sequence dependency is a phenomenon with a significant impact in terms of solution performance to production planning problem, and has been investigated by numerous works such as [3–5]. Sequence dependent setup times depend on the characteristics of the adjacent scheduling units. Independent setups, as expected, do not vary based on the predecessor unit but on the very unit itself. Regardless of whether the setup type is sequence dependent or independent, we can further categorize setups as either product or family setups. In family setups, products are grouped into families with respect to certain attributes affecting the setup time, and setups arise between production units belonging to different families.

In terms of mathematical formulations, we can further investigate setups under two sub domains. First one concerns discrete time formulations where planning horizon is divided into periods. There are two concepts referred to in the literature related, namely carryover and crossover. We refer to [6] to note their difference. In setup crossover, the time spent for the actual setup task can span over two periods including the period boundaries. On the other hand, setup carryover allows a setup state to be maintained from one period to the next one. Second one concerns again discrete or hybrid models with detailed relation to sequence of setup groups, either product or sequence oriented formulations. In the latter, pre-defined setup sequences are allocated to periods whereas in the former models assign products periods.

[7] defines minor setup such as time incurred on machines of moderate length due to switch from one part to another and major setup of long length due to a switch between parts belonging to different families. However, in float glass manufacturing major and minor setups are both related to family setups, former being related to a change in color whereas latter related to a change in coating or thickness. [8] develop a classification scheme for setups, however including cleanings, which can be a key cost driver in process industries such as food and pharmaceuticals. Cleanings can be viewed as another setup type, required due to quality and safety considerations. The paper presents a mathematical model to accurately represent cleanings.

In float glass manufacturing, cleanings similarly translate into setups since switching from one color to another requires the furnace and the molten solution to be stabilized in terms of quality of the destination color. Finally, startup is another category of setup, which corresponds to resources spent to start producing any product. However, we do not further elaborate on details of startup setups, since in float glass manufacturing lines operate on a 24/7 basis and startup setups practically only exist when a new production line starts operating for the first time.

Co-production is producing several different products in a single production run by necessity [9] , and it exists in various industries including petroleum, semiconductor, glass etc. Main difference of co-production from by-production is that co-products are primary products themselves and that a certain combination of products needs to be produced conforming to the necessities of the process along with intended products. On the other hand, a by-product is not primarily produced itself but rather produced as a result of producing another product. Co-production needs to be dealt with in process industries since it can result in undesired production, which means production and inventory holding costs incurred unintentionally.

1.1. Thesis Contribution

Capacitated Lot Sizing Problem (CLP) is a basic production planning problem, and is known to be NP-hard [10]. Moreover, finding a feasible solution for a single-level General Lot Sizing Problem (GLSP), which is single-level special case of General Lot Sizing Problem for Multiple Production Stages, is NP-complete [11]. Our main line of research is on designing efficient solution strategies for campaign planning in the presence of sequence dependent family setups and co-production in different planning complexity levels in terms of number of machines, alternative selection and network structure. We will concentrate on float glass production as application domains.

We defined campaign planning as determination of the campaigns, which translates into timings and durations of production runs of product families.

In the existence of co-production in addition to sequence dependent family setups, campaign planning becomes critical in process industries since the production capacity depends highly on the mix of products allocated, which is not fully controllable. Float glass manufacturing is a process having sequence dependent family setups and co-production attributes. Furthermore, it is a continuous process and the furnace needs to operate 24/7 until it reaches the end of its lifetime.

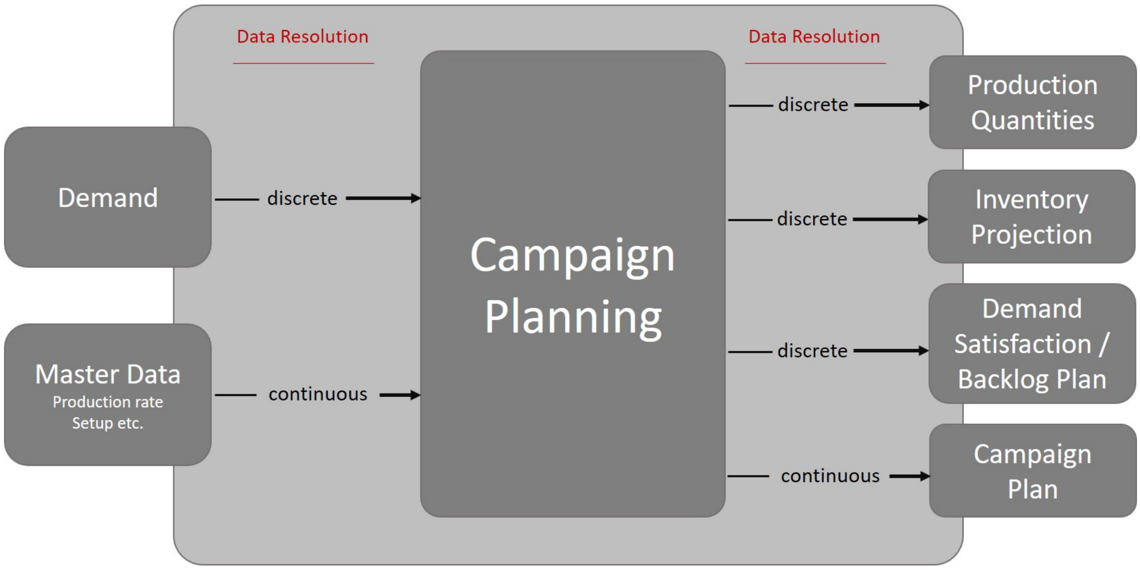

Figure 1.1. Main components of campaign planning problem.

Figure 1.1 illustrates the main components of the campaign planning problem. Tactical planning in float glass manufacturing is typically executed by the planning specialists implementing a manually pre-determined campaign plan. The tactical plans are generated at a monthly level since the demand forecasts are available on discrete time with monthly availability. However, critical information that drives planning activities such as setup durations and production speed is available in continuous time. Hence, the campaign planning problem needs to incorporate continuous timeline while ensuring the demand responsiveness on discrete time. Consequently, the synchronization between discrete and continuous information is challenging. To the best of our knowledge a work that can efficiently incorporate continuous time input data with discrete time data without harming the optimality due to discretization is not present in the literature for different complexity levels.

The main output of the planning is the campaign plan, which we can define as a sequence of families, start and end times of setups and productions. The planning process yields production quantity per product and period based on campaign decisions, which in return provides demand satisfaction and backlog plan as well as inventory projection. Despite focusing on float glass manufacturing as the specific application domain, we believe that the methods we present can be generalized to other process industries.

1.2. Thesis Outline

We present literature review in Chapter 2, from which we observe that former studies can be categorized with respect to the structure of the machine settings. High portion of the studies focus on single machine instance. Other studies cover parallel machine and multiple product hierarchy and multiple facility settings.

We define the details and the challenges of the campaign planning problem in float glass manufacturing in Chapter 3. Availability of input data in both discrete and continuous time requires any solution method to manage the resulting complexity in addition to setups and co-production.

In Chapter 4, we focus on single machine instance of the problem. We propose two Mixed Integer Programs (MIP). Both formulations determines the sequence of product families, which we name as patterns,to be produced on each machine in each period, and both formulations contain macro periods. They are different from each other on formulating setups over period boundaries. One couples all pattern combinations over period boundaries whereas the other couples families.

Chapter 5 considers parallel machine extension. The extended problem considers both identical and unrelated machines. Formulations proposed for single machine problem fail with the new scope. We develop a compact reformulation which leads to Column Generation (CG) and a novel Branch-and-Price (B&P) algorithm.

In Chapter 6, we study the problem in the most general form. We introduce a hierarchy of multiple product levels, which contains both discrete and continuous products. This brings addition of discrete production lines, outputs of which consume outputs of continuous production lines. Moreover, we introduce multiple facilities with one or more production lines as well as customer locations, which hold the customer demands. All of the previous solution methods prove to be not sufficient enough in terms of solution quality in a reasonable running time. As a result, we build mathematical programming based heuristics, namely matheuristics making use of reducing the problem to multiple facility parallel machine instance exploiting the business insights.

In Chapter 7, we present numerical results of proposed solution methods for all the instances of various dimensions in terms of family structure, number of production lines and planning horizon. Finally, Chapter 8 concludes the dissertation with a summary of the proposed methodologies and discusses further research directions.

2. LITERATURE REVIEW

Production planning with setup considerations is a widely studied topic in the literature. Consequently, comprehensive reviews on the topic are available. [12] gives definitions for GLSP, Capacitated Lot Sizing with Sequence-Dependent-Setup, Proportional Lot Sizing and Scheduling, Continuous Lot Sizing and Scheduling and Discrete Lot Sizing and Scheduling. Authors categorize the reviewed papers with respect to being extension to one of these fundamental models. They state that large-bucket models dominate the small-bucket models. In large-bucket models typically multiple items can be produced within a period whereas in small-bucket models at most one item is produced in a period. Authors also note that studies containing multi-stage models are limited to only two production stages and the reason behind is only one or two processes are argued to be bottleneck in real world problems.

Another review study is available in [13]. It categorizes the literature based on shop environment type including single machine, parallel machines and flow shops, batch and non-batch setup indications and sequence dependency. Authors state that the majority of 300 papers they reviewed address sequence independent setups. Branch-and-Bound (B&B) based algorithms, MIP based matheuristics, Dynamic Programming (DP) and some meta-heuristics are the most common solution methodologies used. Authors also suggest that even though most of the current available work is limited to planning activities in manufacturing, models with setup consideration have great potential to be applied to other areas such as telecommunications, logistics and high-speed parallel computing.

An updated version of [13] is in [14] with a review of around 500 papers. The categorization of the papers is the same and this newer version covers again problems involving static, dynamic, deterministic and stochastic environments for different shop types. Authors state that the research on scheduling problems with setup presence is less than 10 percent of the available literature on scheduling and more research is needed.

Another important conclusion is that, around 75 percent of the available papers related to single machine environment with family setup address sequence independent setup type. Hence, there is need for addressing the sequence dependent scheduling problems in single machine environments with family setup times.

Remainder of this Chapter refers to existing work from literature under four sections. First section focuses on studies on single machine case while second one focuses on parallel machine case in a single facility. Section 2.3 discusses studies with network structure and multiple product hierarchy. Finally, we present papers relevant to campaign planning problem in process industries in Section 2.4 and provide a high level comparison of our study with existing work. At the end of this Chapter, Table 2.2 provides an overview of the available literature and the acronyms used within is available in Table 2.1.

2.1. Single Machine

In the existence of a pre-defined jobs heuristic algorithms are applied frequently in order to determine the sequence forming up the final schedule. In [15], the problem consists of a set of jobs, where each belongs to a family, in continuous casting stage of steel industry. Sequence-dependent setup is required if consecutive jobs are from different families. Material constraint exists in the form of cumulative demand for an upstream resource restricted to its cumulative demand assuming linear supply and consumption. Authors apply Variable Neighborhood Search (VNS) with 6 different moves, namely job move, job exchange, batch move, batch exchange, batch combine, batch break. At each iteration, the algorithm selects a move and explores all neighbors unless one provides an incumbent solution. Otherwise perturbation is done based on a score value calculated for each job. Critical jobs are decisive for the move.

Authors of [16] claim that traditional position change heuristics are inefficient as neighborhoods contain a lot of non-improving solutions. For a problem similar to [15] in terms of jobs and families, they propose a batch-based Simulated Annealing (SA).

The algorithm has a neighborhood definition aimed to increase efficiency in neighbor detection by eliminating non-promising neighbors. Proposed neighborhood is based on batch destruction and the schedule gets completed again with a greedy heuristic.

In [17] the authors study non sequence dependent but family dependent job scheduling without preemption. Six different heuristics, Incomplete Dynamic Programming (IDP), Earliest Due Date (EDD), Rolling Horizon (RH), Group Technology (GT), Local Search Modified EDD and Batch Splitting GT, are compared and RH performs the best out of 1440 randomly generated test instances. In [18], the problem contains sequence dependent product setups and proposed solution is based on Scatter Search with motivation to develop a new meta-heuristic that will provide near-optimal solutions within a reasonable amount of time. Improvement module is based on two different VNS based local search and diversification is based on both random and construction heuristics. Moreover, a reference set strategy is employed. Search starts with a small size which then is increased if no improvement obtained in certain number of periods. Authors of [15–18] employ a different heuristic strategy to get a good solution within a reasonable amount of time.

In [19] capacitated lot sizing problem under sequence independent setup is formulated as Mixed Integer Linear Program (MILP) with setup carryover. Fundamental decision of setup carryover is formulated with a binary variable to indicate whether a setup is carried over to its adjacent period. Authors also define another binary variable indicating whether a setup state covers a period entirely. Since MILP formulation is unable to solve the problem, they propose a Genetic Algorithm (GA), which they argue has difficulty in finding exact minimum/maximum optima in a large and complex solution space. To tackle this, they propose a hybridization approach: GA for locating good quality solutions and fix-and-optimize to search this region more in detail. It is also important to note here that they apply time based decomposition to the original MILP formulation to generate initial solutions with a mixture of partially random information and partially information from Linear Program (LP) relaxation.

Authors study lot sizing on a single machine, single level and multiple-product with product dependent setup in [6]. Main contribution of the work is that setup crossover between periods is now possible without adding binary variables. Another contribution is that model contains symmetry breaking constraints. They propose two formulations for the problem. Binary variables indicating whether a setup is split between periods and continuous setup borrow time variables are important to notice in the first formulation. In the second formulation, binary variable set indicating whether a setup is complete in period t and another set indicating whether a setup is split in addition to continuous setup back and front variables corresponding to time spent for setup in two adjacent periods respectively. Authors show that binary variables in first formulation can be relaxed and symmetry breaking constraints improve performance in second formulation whereas barely have effect in first.

The work in [20] focuses on lot sizing problem with sequence dependent setups. They argue that it's not possible to solve to optimality in reasonable time and hence the usage of heuristic methods are necessary. Variables to keep track of demand satisfactions and flow equations and elimination constraints have positive impact for a tighter lower bound. In formulation, macro periods correspond to days whereas micro periods correspond to parts of a day. In order to schedule the first day more in detail it has 10 micro periods while the rest of the days have a single micro period. they apply Relax-and-Fix (R&F) heuristic as the solution procedure. First, all binary variables other than first day's are relaxed and solved. Then they're fixed and rest of the binary variables are restored. First step is argued to be slow and they apply Descent Heuristic (DH), Diminishing Neighbourhood Search (DNS) and SA metaheuristics.

The work [21] studies animal feed compound production, where some products serve for cleansing as long as they are produced a certain amount, which results in violation of triangular inequality of sequence-dependent setups. They apply a R&F heuristic, which is shown to be computationally and economically effective compared to current practice in the industry. [22] study production planning for animal nutrition products under sequence-dependent family setups, and formulate a mathematical model.

The model is based on asymmetric traveling salesman problem. The study shows the model can be efficient for some certain cases but needs further algorithmic development for variants of the problem.

The authors present another study based on setup carryover concept with sequence dependent setup times and costs in [5]. The formulation is sequence oriented, which means that for a period t a pre-defined setup sequence is selected and assigned. The authors propose an efficient and fast sequence enumeration method. Rescheduling is basically limiting sequences that can be selected for a given period with respect to fixed jobs. Moreover, a lower bound generation scheme is proposed, result is used to prune the search tree.

In [23] a change of paradigm is proposed in lot sizing and scheduling named block planning concept. It is based on a continuous representation of time. A block is actually a timespan in which a setup family can be scheduled and a setup family is a pre-defined sequence of products between families resulting in major setup and assigned to blocks with a binary variable. Model avoids overlapping blocks via block start-finish variables and corresponding constraints. Authors also argue that as setup and inventory holding costs are hard to determine in most practical cases, timespan minimization is a reasonable objective.

Uncapacitated dynamic lot-sizing problem with co-production extension is studied in [9]. Co-products' demands are non substitutable and co-production is always approximated with random yield. However, it is also argued that the yield can also be determined as a percentage of total production with respect to historical data. Furthermore, co-production is defined to be 1-to-1 between products whereas by-production is defined to be 1-to-n with respect to certain percentages. Author proves that the problem can be reduced to single item lot sizing problem and hence DP is suitable.

Short-term production planning and scheduling in pulp and paper industry is studied in [24]. The main objective is maximizing production throughput.

Consequently, planning should minimize the losses due to sequence dependent setups. This is usually the case in process industry. It is also important to note that the Bill-of-Material (BOM) hierarchy consists of two-stage. Digester speed is determined with a binary variable assigned per micro period among a known set of speed levels. Between sub-periods there is a maximum allowed variation in speed of digester and as a result model includes coupling constraints. Setups are due to paper grades, which are product families that require setup. Solution representation for the hybrid VNS is based on setup sequence and the procedure itself is a combination of exact method, for continuous variables, and Speeds Constraint Heuristic, for digester speed. [25] also formulate a MIP model for lot scheduling in pulp and paper industry in integrated mills. They propose a GA to efficiently solve large instances.

Authors of [26] study the same problem again in pulp and paper industry to the extent of development of a decision support system. Production campaigns correspond to paper grade and the aim is to determine their size and sequence. Formulation contains macro and micro periods. Time slots, micro periods namely, are of variable length and independent from macro periods which have the demand information. The model has two sets of binary variables. Minimum and maximum duration constraints for campaigns are present in addition to a soft constraint for encouraging minimum time between two similar campaigns. As solution strategy, the paper uses a meta-heuristic which is composed of 3 stages, initial solution, forward pass and neighborhood search. Heuristic is able to provide satisfactory results in reasonable running times.

Authors of [27] study the same problem in glass container industry. Glass color, which causes a major setup in glass manufacturing environments, is assumed to remain constant in short term. Sequence dependent setup is due to product changeover. Furnace, which needs to be working 24/7, capacity is not formulated as equality but the model penalizes unutilized capacity. Heuristic applies multi-population GA assuming convergence and then applies SA to the incumbent solution to intensify the search. To determine the number of mold cavity a MIP is formulated.

2.2. Parallel Machine

Production planning and scheduling problem with lot sizing has an important dimension, namely alternative selection, when multiple machines are present. [28] studies the problem for short term in glass container industry with identical parallel machines. study argues that production losses due to not using all of available capacity of a resource is critical in process industries which glass container is of one, and this paper claims to be the first to address this issue. Model penalizes production losses, and decides integer number of mold cavities. The solution includes relaxation is based on these variables. Moreover, model contains valid inequalities to improve the quality of lower bounds. One important observation is that impact of the inequalities increases as the number of products and periods increase. In relaxation reformulation, product assignments are eliminated with the setup carryover constraints. The problem then reduces to a network flow representation. Production quantities can be determined by using a shortest path algorithm, which ensures integrality of setup changeover variables directly.

Authors use hierarchical approach in food industry in [29]. The importance in the supply chain coordination is high due to perishable products, since products should be shipped as soon as they're produced. Hierarchy consists of batching of orders, production planning and finally distribution sub-problem. Similar customer orders are grouped to form batches with due dates. Production planning then schedules these batches with sequence dependent setups with a MILP. Distribution planning minimizes trade-off between transportation cost and quality decay. The paper compares hierarchical planning approach to integrated planning and concludes that a certain level of quality can be guaranteed without increasing costs too much.

Authors of [30] study production planning and scheduling in oil refineries concentrating on subsystems due the complex nature of the planning problem. In practice, refineries develop in-house developed simulations based planning tools. A complex Mixed Integer Non-Linear Program (MINLP) model is described in two parts.

First, operating rules are formulated with binary variables in order to ensure the hard constraints of the working mechanism of subsystems. Second part consists of material flow constraints that are basically the part, where stream flow rates and viscosity decisions are taken. Viscosity constraints cause co-production and non-linearity but then are reformulated to obtain an MILP formulation. As per co-production, types and rates of co-products are known in advance. The most difficult step throughout the entire study is to understand working mechanism of the refinery, which is obvious to the refinery experts but unclear to the planning system researchers.

In [31], authors present several procedures for scheduling identical parallel machines with family setups minimizing total tardiness. It applies Tabu Search (TS) with batch insertion move. In case of no improvement, job with highest tardiness is split and the procedure continues until the stopping criteria is met. Authors also apply GA with n-tournament selection operator and uniform order-based crossover. It uses shift mutation operator and applies local search to intensify the search. Finally, an optimal branch-and-bound algorithm with implicit complete enumeration is applied. Computational tests show that GA performs best among all proposed procedures and in small instances finds the optimal solution in most cases.

Authors of [32] study scheduling of elective surgeries to multiple operating rooms, a different domain than production planning. Elective surgeries can be scheduled as opposed to urgent surgeries as they do not stand emergency and there are types of surgeries each having different requirements for operating rooms. This results in sequence dependent setup times between surgeries. Decisions to be made are number of operating rooms to open, assignment of surgeries to operating rooms and sequence of surgeries within an operating room. Paper proposes MINLP and Constraint Programming (CP) formulations and authors show that CP outperforms MINLP model as MINLP gets inefficient with the increase of problem size.

Five novel MIP formulations are proposed for identical parallel machine scheduling with family dependent setups in [33]. The formulations are inspired by single commodity, arc-flow and set covering formulations. They conduct extensive set of numerical experiments and show the efficiency of two of the formulations driven from strong bounds bounds. [34] study the same problem on single machine. They formulate a model exploiting properties of optimal solutions. They discuss the LP relaxation of their formulation to be stronger than other formulations in the literature. Hence, the model is able to find optimal solutions for instances with high number of families and long setup times.

Authors of [35] study capacitated lot sizing and scheduling problem with alternative selection, and the problem is generalized to parallel machine case. A stochastic MIP based decomposition heuristic is the proposed solution approach. As the heuristic proposed is an improvement heuristic, algorithm consequently requires a feasible solution. Authors use a construction heuristic based on the general R&F framework. It does not apply relaxation on integrality of any integer variable since the neighborhood is defined as a subset of adjacent periods and products. This means optimizing all the related variables while fixing setup variables of other periods and products.

Three formulations (discrete, hybrid and continuous) are proposed in [36] for two stage lot sizing and scheduling problems with continuous upstream and discrete downstream production that is present in many process industries. In discrete model, planning horizon is divided into macro periods which are then divided into micro periods. Number of micro periods is assumed to be user defined and their length is variable. Hybrid model is batch scheduling within periods with upper bound on the number of batches that can be scheduled within the planning horizon. Finally, continuous model works with an initial number of available common resource batches with unknown sequence. The work concludes that continuous model is the most flexible allowing setup crossover but has bad performance. Discrete is favorable being compact and providing good solutions yet without optimality proof.

Hierarchical production planning has some problems such as potential infeasibility in capacity and setup times due to aggregation and hence its optimality becomes doubtful as well as cost savings results. The purpose of study in [37] is to solve such problems with an integrated approach. Main decisions in the problem formulation contains workforce hiring and firing, subcontracting, storage and backorder, and sequence dependent setups. Sequence of setups are modeled with a set of binary variables indicating its position within a period. Quantity of a setup family needs to be in between a certain percentage of the production of the family, which can result in co-production with demand being less than the minimum required production amount. Integrated model outperforms hierarchical approach in terms of setup number and overall costs. However, it can obtain with more than 20 hours of running time.

A new heuristic method Hamming Oriented Partition Search based on mathematical programming is proposed by [38], to solve the lot sizing and scheduling problem in textile industry consisting of two stage process each of which can either be executed on single or parallel machines with alternative selection. Yarns are grouped into families which are then related to fiber blends. This ensures their quality and yarn families have sequence dependent setups. HOPS consists of B&B combined with a problem-oriented procedure injecting new and better upper bounds into the original problem. The heuristic is shaped with respect to problem features, for determining set of variables to be fixed. This partitioning is based on a metric such that variables that are the most promising partition to be optimized are determined. Another key feature is that previous solutions are stored with a coefficient regarding recency and these are used in partition determination in order not to re-optimize stable variables.

Production planning problem of a wood remanufacturing mill with following characteristics is studied in [39]: co-production that is uncontrolled in most cases, alternative selection, Make-to-Order (MTO) with short customer lead times, sequence dependent family setup and finite capacity. Sophisticated setup formulation for exactly four product families. The model has a set of binary variables for individual families and another set for indicating number of families scheduled in a period.

The objective is backorder minimization, which is the characteristic of the industry regarding the MTO and short lead time nature. The paper proposes an efficient re-planning based two phase solution procedure, and it is shown with simulation based tests that proposed solution improves backorder performance.

Authors of [40] propose an improvement heuristic based on Variable Neighbourhood Decomposition Search and fix-and-optimize to solve general multi-level lot-sizing and scheduling problems. Sequence dependent setup crossover between period boundaries is possible with setup back and front variables representing time spent for setup in predecessor and successor periods respectively. They sequentially apply three decomposition schemes based on product, resource and process, and procedure can outperform a commercial solver in small instances but gets producing worse optimality gaps for real world instances within one hour of running time.

Glass container production contains sequence dependent family setup times based on color of the glass produced by furnace, which is the first production stage. The main aim of [3] is to determine the color campaign schedule. Products belong to color families and products with same color within the same family do not require a setup for a changeover. Similar to [27], continuous production with 24/7 uptime is declared but capacity not utilized is penalized, hence not guaranteed. For the initial solution generation, authors propose a construction heuristic, which consists of product selection with respect to five criteria and scheduling. Proposed heuristic is a combination of Reduced Variable Neighborhood Search (RVNS) and basic VNS with RVNS aimed to increase efficiency whereas VNS to balance the effectiveness. According to the comparison between pure VNS, pure RVNS and RVNS/VNS, the proposed variant, VNS is shown to be superior to RVNS and the variant. However, due to a worse initial solution and short solution time, available proposed methodology becomes more attractive.

Authors of [41] study planning and scheduling of drying and finishing operations in softwood lumber industry with two characteristics, divergent system with co-production and alternative production process. All different combination of processes are reduced down to alternatives, which are then assigned to machines per period. It is also important to note that a product can be both consumed internally or sold. Authors propose MIP and CP formulations. Tests show that performance of MIP is unstable with respect to dataset due to large number of binary variables while CP provides good quality solutions fast with proposed search strategy.

In [42], the study focuses on generalization of lot scheduling problem including backordering and setup carryover on unrelated parallel machines. They formulate three different matheuristics inspired by local search, local branching and feasibility pump. Their tests show that their approach outperforms other approaches and two MIP solvers on base formulation.

The aim is to minimize total weighted tardiness for scheduling unrelated parallel machine scheduling problem with sequence dependent setup times and machine eligibility restrictions in [43]. They propose a SA and a TS algorithm. Numerical experiments show TS with long-term memory yields better solutions.

2.3. Multiple Level Network

Having multiple facilities including production sites, warehouses and point of sales increases the complexity of the problem as the decisions to be taken also include allocation of campaigns to facilities in addition to already defined sequencing and lot sizing related decisions presented in previous sections. Allocation to facilities become more important when the distribution within the network to customer locations is also to be planned.

Authors of [44] study production planning problem in biopharmaceutical processes with multiple BOM levels. The industry has its own characteristics such as batch and continuous processes being present at the same time, multiple intermediate deliveries, sequence dependent setup and product shelf life limitations. MILP is based on resource Task Network with continuous time formulation with a single time grid that is structured with sub periods, namely event points, the end times of which are determined through a decision variable resulting in variable sized periods. Authors introduce a limitation on number event points and lot scheduling coupling such as maximum number of points a lot can traverse. This last might be problematic because period length is variable and model only limits by the number of periods which can basically be either too long or short.

The study [45] focuses on discrete time MILP formulation for lot sizing and scheduling with multiple BOM levels and sequence dependent setup times. Planning horizon is divided into macro periods attached to due dates and micro periods attached to campaign allocation. Carryover typed setup changeovers are tracked with a linearized variables. An important note about campaign allocation is that only the latest campaign can span multiple slots. Maintenance and product trials are also introduced as tasks to the system such that they're unavailable time within a period. Proposed strategy couldn't provide better solutions than 10% relative gap.

In [4], authors also studies glass production is studied also, this time in glass container sub domain. Main objective of the study is to determine color campaigns in furnaces in different plants, and synchronize the overall operation. Campaign duration is a decision variable and not necessarily integral as number of days. Moreover, model also takes a near-strategic decision as furnace shut-down, after which the furnace can not start up again. Demands do not have to be entirely satisfied which introduces unsatisfaction penalty. As initial formulation is hard to solve, the study proposes a relaxation, and solution with an improvement heuristic that is a combination of multi-population GA and fix-and-optimize. Proposed heuristic outperforms a commercial solver in most instances.

Authors of [46] work on a production distribution formulation in network context in another process industry, metals. There are several sites with one or more lines and capacitated warehouses for tackling fluctuating demand. Sixteen products are divided into two families that require sequence dependent setups for a changeover in between. Study employs single setup per period assumption and this is justified denoting the possibility of having nine days of setup within a month if more than two families are selected to be produced due to their demand patterns. In a capital intensive process industry like metal, nine days without output is argued to be unacceptable. As only two families are present, all possible sequences are enumerated completely and the formulation is hence sequence oriented. Proposed model is implemented as a decision support tool but is not capable of solving large problems.

Authors of [47] study tactical production and distribution model with continuous first and discrete second stage production process. Network structure contains plants, warehouses and customer sites having demands. As opposed to [46], warehouse capacities are assumed to be sufficiently large here. Two formulations are proposed as MILP. First one is sequence oriented in which pre-defined sequences are allocated to resource periods. Second one is product oriented specifically determining which families, at most two are allowed, are produced in a resource period. R&F construction heuristic with disjunctive subsets and groups by period is applied in addition to fix-and-optimize heuristic with an adaptive VNS. The study defines neighborhood on a subset of periods and resources.

2.4. Relevant Campaign Planning Studies in Process Industries

Authors of [2] deal with tactical level production planning problem in float glass manufacturing company that has four facilities distant from each other with significant transportation cost. Glass manufacturing has its own unique properties, continuous production with random yield, partially controllable co-production, product substitution and complex sequence dependent setups to name a few. However, in the formulation color campaigns are assumed to be determined beforehand as inputs.

Moreover, setups due to thickness changeover are neglected. Co-production driven by size group, quality and stacker capacities, on the other hand, is formulated with a special set of constraints. Although resulting setups are neglected, thickness changeover phenomenon is modelled with base thickness assumption meaning meaning base thickness will most probably be produced in each period as by definition it contains most common products. With proposed system more detailed production plans can be generated respecting size/quality restrictions, improved production quantity and inventory levels, decreased transportation costs with faster plan cycles for the planners.

The work in [48] is on lot sizing and scheduling problem with sequence dependent setup consideration on single machine with two different approaches. One formulation is based on decisions for setup between products whereas in the other uses a collection of pre-defined sequences. The latter selects a sequence to be executed in the production. However, authors do not explicitly model family setups but only products and longer setups between products corresponding to family aggregation is not analyzed in detail but only present in a single instance of computational experiments. Moreover, the models proposed do not allow for setup crossover, which can be necessary in environments where some of input data is not an integer multiple of micro-period lengths such as setup durations.

In [49], the authors study extensively the float glass manufacturing process and develops a MIP for production planning. The model in this study determines whether a product is produced in a time period and that at most one product is allocated to periods. The author does not explicitly address sequence dependent family setup phenomenon. [50] model the transition between adjacent periods permitting the changeovers between products occur before, across and after the period boundaries. However, fixed number of slots, similar to micro-periods in [48], can result in suboptimal solutions in cases where input data is sensitive to discretization.

The authors of [51] study the effect of uncontrolled co-production on the production schedules and the environment contains multiple products with unsubstitutable demand in glass manufacturing. Moreover, the problem is based on uncapacitated single machine instance. Co-production is not scrap, has its own demand and there is product hierarchy with respect to quality and size attributes. Moreover, demand and co-product rates are deterministic. Common cycle schedule method is applied. Authors found out that cycle length increases with increasing co-production rate but sensitivity of the long-run average cost to the co-product rate is low.

In [52], products are aggregated to product families, which have sequence dependent setups in between. Products within the same family have sequence dependent setup times. MILP formulation contains an excessive number of binary variables including family allocation per period, product allocation per period and sets indicating whether a product is first or last in family allocation per period. Moreover, setup crossover between periods is possible with setup back and front variables. As also discussed in [28], not using all of available capacity is not desired in process industry and authors introduce dummy product to let capacity constraints be equality. However, it is not clear how this is discouraged within the formulation.

To summarize, there are studies in the literature for campaign planning problem in process industries. However, the available work do not simultaneously solve campaign planning with setup carryover considerations, does not provide a solution for the synchronization of discrete and continous input data, or have some certain limitations, which may produce sub-optimal solutions in some environments. Moreover, 24/7 working mode of resources are either relaxed or relaxed with penalty on unusued capacity. Hence, we believe our work will provide valuable contribution to the literature.

Table 2.1. Acronyms for problem attributes.

Columns	**Potential Values**	**Acronym**
Network Structure, Alternative Selection, Lot Sizing	Yes	y
	No	n
Facility, Machine	Single	s
	Multiple	m
BOM Level	Single	1
	Two	2
	Multiple	M
Setup Type	Sequence Independent	si
	Family Dependent	fd
	Product Dependent	pd
	Sequence Dependent	sdp
	Sequence Dependent Family	sdf
Setup Formulatiıon	Sequence Oriented	so
	Single Setup per Period	sspp
	Multiple Setup per Period	mspp
Setup Period Relation	Crossover	xo
	Carryover	co

Table 2.2: Literature overview.

Reference	Network Structure	Facility	Machine	Alternative Selection	BOM Level	Lot Sizing	Setup Type	Formulation Approach	Period Schema	Industry
[44]	n	s	m	y	M	n	sdp	sspp	co	Biopharmaceuticals
[45]	n	s	m	y	M	y	sdp	sspp	co	-
[47]	y	m	m	y	2	y	sdp	sspp	co	Glass container
[36]	n	s	m	y	2	y	sdp	sspp	xo	Spinning
[40]	n	s	m	y	M	y	sdp	sspp	xo	-
[46]	y	m	m	y	1	y	sdp	so	co	Metal
[5]	n	s	s	n	1	y	sdp	so	co	High-tech
[35]	n	s	m	y	1	y	sdp	sspp	co	-
[20]	n	s	s	n	2	y	sdp	sspp	co	-
[23]	n	s	s	n	1	y	sdf	so	–	Beverage
[4]	y	m	m	y	M	y	sdp	sspp	co	Glass container
[3]	n	s	m	y	M	y	sdf	sspp	co	Glass container
[19]	n	s	s	n	1	y	si	sspp	co	-
[29]	n	s	m	n	1	n	si	sspp	co	Perishable food
[28]	n	s	m	n	1	y	sdp	sspp	co	Glass container
[38]	n	s	m	y	2	y	sdp	sspp	co	Textile
[24]	n	s	s	n	2	y	sdp	sspp	co	Pulp & paper
[27]	n	s	s	n	2	y	sdp	sspp	co	Glass container

Table 2.2. Literature overview (cont.)

Reference	Network Structure	Facility	Machine	Alternative Selection	BOM Level	Lot Sizing	Setup Type	Formulation Approach	Period Schema	Industry
[52]	n	s	m	n	1	y	sdf	mspp	xo	Beverage
[26]	n	s	s	n	2	y	sdp	mspp	xo	Pulp & paper
[15]	n	s	s	n	1	n	sdf	–	–	Steel
[31]	n	s	m	n	1	n	sdf	–	–	-
[9]	n	s	–	n	1	y	si	sspp	–	-
[37]	n	s	–	n	1	y	sdp	sspp	co	Mold
[16]	n	s	s	n	1	n	sdf	–	–	Steel
[32]	n	s	m	n	1	n	sdp	–	–	Health care
[17]	n	s	s	n	1	n	fd	–	–	-
[6]	n	s	s	n	1	y	pd	sspp	xo	-
[18]	n	s	s	n	1	n	sdp	–	–	-
[2]	y	m	m	y	1	y	–	–	–	Float glass
[30]	n	s	m	n	1	y	–	–	–	Petroleum
[41]	n	s	m	y	M	n	–	–	–	Lumber
[39]	n	s	m	y	2	y	sdf	mspp	–	Wood
[51]	n	s	–	n	1	y	si	–	–	Glass
[11]	n	s	s	n	1	y	–	–	–	-
[48]	n	s	s	n	1	y	sd	–	–	-
[48]	n	s	s	n	1	y	sd	–	–	-

Table 2.2. Literature overview (cont.)

Reference	**Network Structure**	**Facility**	**Machine**	**Alternative Selection**	**BOM Level**	**Lot Sizing**	**Setup Type**	**Formulation Approach**	**Period Schema**	**Industry**
[50]	n	s	s	n	1	n	sd	–	–	Glass
[21]	n	s	s	n	1	y	sdf	mspp	co	Nutrition
[22]	n	s	s	n	1	y	sdf	sspp	co	Nutrition
[25]	n	s	m	y	2	y	si	mspp	co	Pulp & paper
[42]	n	s	m	y	1	y	sdf	sspp	xo	-
[43]	n	s	m	y	1	n	sdp	–	–	-

3. PROBLEM DEFINITION

3.1. Flat Glass Manufacturing as a Process Industry

Process manufacturing differs from discrete manufacturing in how it creates the final products. Discrete manufacturing considers producing identical products usually over an assembly line, whereas process manufacturing converts raw materials into final products through a continuous process following a recipe [53]. Moreover, the output of a process line cannot be disassembled back to its input. Process industries are capital oriented since there is usually high associated manufacturing and raw material costs. Consequently, the main driver within the manufacturing process is cost effectiveness, which makes the planning activity more intricate.

By nature of supply chain in manufacturing industries, transportation, inventory holding and demand satisfaction related costs are directly considered in planning decisions as previously expressed. Nevertheless, loss of efficiency in capacity usage can have non-negligible implicit impact on the overall effectiveness. The reason is again due to dynamics of most process industries. In glass manufacturing, each furnace need to be up and running 24/7 due to the continuous production nature and energy costs increase as a significant expense item. These can be seen as fixed operating costs, however the lost capacity is highly undesirable. Setup times and, if the nature of the process imposes, co-production certainly need to be dealt with to improve the effectiveness.

Float glass manufacturing is a process industry, where the main driver in planning process is the cost and the effectiveness in capacity usage. Furthermore, float glass manufacturing has some special characteristics making it difficult from a planning complexity perspective, which we discuss next.

3.2. Process Definitions

The term float refers to the physical nature of the glass production. Molten solution, consisting of raw materials such as sand, limestone and soda ash, is fed into a tin bath and transforms into its flat form by floating over liquid tin. The floating glass then goes through a coating process depending on the characteristic of the active production. Please note that such products are not categorized as coated products but a special type of float glass products. This is a chemical process consisting of covering the surface of float glass with thin metal layers [2]. It enhances visual and thermal properties of the final product. Annealing step is where the product cools down and becomes solid. Finally, the glass is cut into different sizes before being picked up and stacked in storage area. Figure 3.1 illustrates a typical float glass furnace and the entire production line.

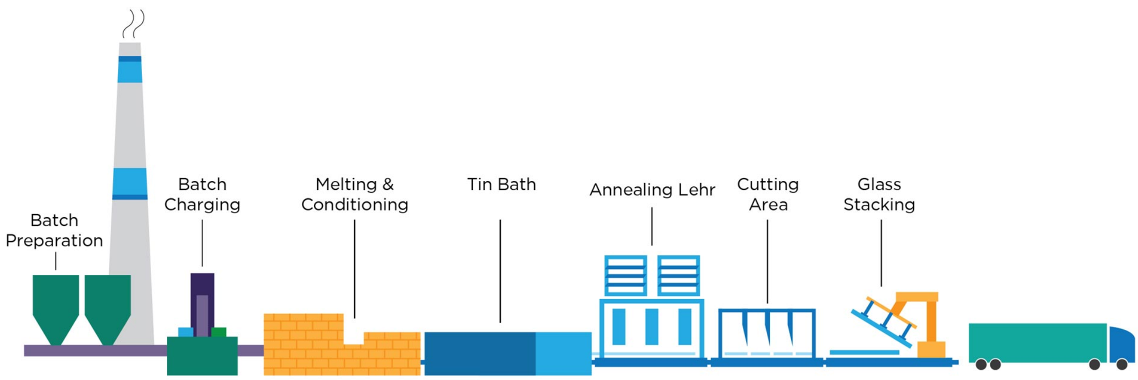

Figure 3.1. An illustrative float glass furnace and production line [1].

The primary characteristics of the finished product is determined by raw materials fed into the mixture [2]. The most significant attribute of float glass is its color. Coating is also another attribute, which has some impact on the production itself. Color and coating identify product family in float glass manufacturing. Products, on the other hand, have size group and quality attributes on top of family attributes.

3.3. Setups

Switching from one color to another requires several days to be spent as setup since the process, the furnace in particular, needs to stabilize to obtain the desired color. Typically, a color changeover takes from three to seven days. This results in significant amount of time and energy consumption without any yield since the production is uninterruptible, hence is very costly. Glass produced during setup time is usually does not have any demand, and needs to be broken into pieces, which then is fed back to the raw material mixture to a limited extent.

In order to compensate the setup cost incurred for the changeover and also for efficiency purposes, each family has a corresponding minimum production duration. Moreover, these setups depend on other family attributes of glass other than color, namely coating. The problem hence contains the phenomenon of sequence-dependent family setups. Other types of setups explained more in detail in Chapter 1, are not relevant in float glass production.

3.4. Co-Production

We define co-production as producing several different products in a single production run by necessity. Due to the chemical nature of the process, random errors on the glass surface appear during production. There can be different types of error such as visually detectable defects. There can also be serious errors which would enforce the output to be scrapped [2]. Depending on where the final product is used, some of these errors can be disregarded. Hence, we can categorize glass with different errors into quality groups. Depending on the cutting decisions regarding the size, the line can yield different size and quality combinations.

Using the historical data that reflects the characteristics of a specific production line, we can determine the percentages up to which a specific combination of products from the same size group and quality group can the furnace yield at most.

For example, producing high quality glass in big sizes on a specific production line might eventually result in an increase in production of moderate and/or low quality glass in lower sizes, which results in uncertainty in final production quantities. However, aggregating production quantities in planning phase can provide enough flexibility to planners since the amount planned is not continuous. It needs to be obtained within a certain timespan. We can define this as partially controllable co-production. For a more detailed explanation on float glass manufacturing fundamentals, we refer to [2].

3.5. Campaign Planning And Challenges

We concentrate on float glass manufacturing for the campaign planning problem, as stated in Chapter 1. Campaign planning in float glass manufacturing needs to deal with sequence dependent family setup, which stems from color of products and co-production, which is due to chemical properties of the process.

Figure 3.2 illustrates an example of a campaign plan for four periods. With the help of this illustration, we can observe synchronization of input and output data, which are available on different time resolutions. We focus on a specific product from family FM. For each period, a production amount and demand for the product is available. On the other hand, the campaign plan is available on continuous time. For example, a campaign of family FM starts in Period 1 and ends in Period 2. Production quantity within this campaign is associated with Period 1 and Period 2 with respect to time overlapping with each one of them. As a result, the production quantity is disaggregated to discrete time. With the help of the dotted lines, we can also observe the illustration of demand satisfaction schema. For example, the demand of Period 2 is satisfied from productions in Period 1 and Period 2. whereas the demand of Period 3 is partially satisfied from Period 2 and Period 4, which results in backlogging. Moreover, for each period, considering the production quantity and demand satisfaction plan, one can obtain ending inventory projections.

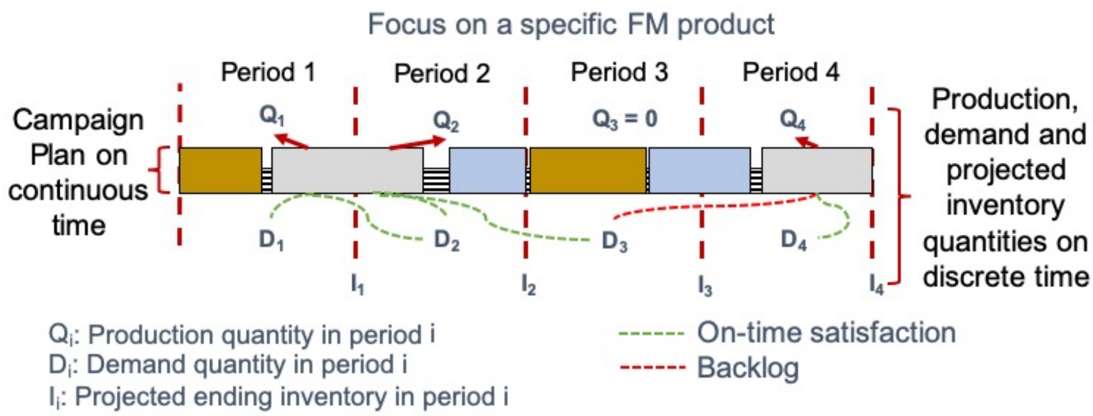

Figure 3.2. Illustration for inputs and outputs of Campaign Planning.

Let us note the main characteristics of the campaign planning problem as follows:

- Demand forecast per product is available on a discrete time (monthly).
- Input master data consists of inventory holding cost, demand backlog and unsatisfaction cost, production speed per item and setup duration between families. They are parameters of the decision process and are available in continuous time.
- Main cost items are inventory holding, demand backlog/unsatisfaction and setup. Production costs are ignored since the problem is on a single machine.
- Setups are costly such that the furnace consumes as much as energy as in production without yielding any glass in order of days in duration. Hence, setups are important in terms of ensuring cost effectiveness of the plan.
- Due to significant setup duration and costs, campaigns are encouraged to have relatively long durations. However, since this will also effect the demand satisfaction plan. Backlog is another major expense item. Hence, obtaining an good quality, or optimal if possible, campaign plan is crucial.
- Due to the fact that sequence-dependent setup times are expressed in continuous time, the campaign plan needs to be on continuous time.

To elaborate on the last item, we note that campaign planning problem differs from aggregate planning even though lot sizing decisions need similarly to be in discrete time in order to match the availability of the demand forecast.

On the other hand, as stated sequencing decisions considering sequence-dependent setup times and production speed is in continuous time. Hence, synchronizing discrete and continuous information is a necessity for the effectiveness of the final campaign plan.

4. SINGLE MACHINE PROBLEM

In this chapter, we develop two mathematical models for the campaign planning problem described in Chapter 3 on a single machine. Both models are mainly based on the state decisions of the machine in each time bucket and they mainly differ from each other with respect to the formulation of the state transition over period boundaries. We name the models Pattern Transition Based Model (PTBM) and Family Transition Based model (FTBM) respectively.

In order to clarify the formulations, we first define the concept of pattern in Section 4.1 as well as the approach for generation, and then introduce the formulations in Sections 4.2 and 4.3. In addition, Table 4.1 illustrates symbols used in both PTBM and FTBM.

Table 4.1. Symbols used in both formulations.

Set	**Description**
J	Set of products
Q	Set of quality groups
S	Set of size groups
T	Set of time periods
P	Set of campaign patterns
F	Set of product families
O	Set of orders for timing of production in a period (b: beginning, m: middle, e: end)
$P(f)$	Set of patterns containing family f at least once
$F(p)$	Set of families belonging to pattern p
$F^o(p)$	Set of families appearing in order o in pattern p
$P^o(f)$	Set of patterns containing family f in order o
$J(f)$	Set of products belonging to family f
$\Gamma(f, g)$	Set of product family couples that are infeasible, $f, g \in F$

Table 4.1. Symbols used in both formulations. (cont.)

Parameter	**Description**
D_{jt}	Demand of product j in period t
$I_{j(-1)}$	Beginning inventory of product j
v_j	Production speed of product j, machine-days required for unit production
A_t	Available capacity of the machine in period t in days
$S(j)$	Index of the size group of product j
$Q(j)$	Index of the quality group of product j
R_{fqs}	Maximum production ratio/percentage for quality group q and size group s for family f
MD_f	Minimum production duration for family f in days
NT_{fp}	Number of times family f appears in the middle order of pattern p
MD_{fp}	Minimum production duration for family f in middle order of pattern p, can similarly be expressed as $MD_f NT_{fp}$
ST_p	Setup time needed for family order within pattern p in days
ST_{fg}	Setup time needed for switching from product family f to family g in days
h_j	Inventory holding cost for product j
b_j	Cost of backlogging a demand of product j for a single period
c_{fg}	Setup cost of switching from family f to family g
c_p	Total setup cost of family order within pattern p
Variable	**Description**
I_{jt}	Inventory of product j at the end of period t
S_{jtk}	Satisfied quantity of demand from period t of product j in period k
U_{jt}	Unsatisfied quantity of demand from period t of product j
X_{jt}	Production quantity of product j in period t
δ_{pt}	Binary indicator variable for selection of pattern p in period t
d^o_{ft}	Number of days spent for production of family f in order o in period t

4.1. Pattern

4.1.1. Definition

We can define a pattern as an ordered list of families that will be produced consecutively within a period. The concept of pattern is similar to sequence in [48] with the difference that they define sequence by product order but we define patterns by family order.

An important issue to address in pattern definition is that setup times are respected. Each adjacent pair within the pattern needs to be feasible in terms of setup changeover. Let FM, MV and BR be three families available. We can define Pattern 1, a pattern with single family as FM, Pattern 2, a pattern with two families FM-MV, Pattern 3, a pattern with three families BR-MV-FM and Pattern 4, another pattern with three families MV-BR-MV. Figure 4.1 illustrates these four example patterns. Notice that these represent sequence of the families that the furnace will produce in a period. In addition, the setup from family FM to MV (for Pattern 2), BR to MV, MV to FM (for Pattern 3), MV to BR and BR to MV (for Pattern 4) should be feasible.

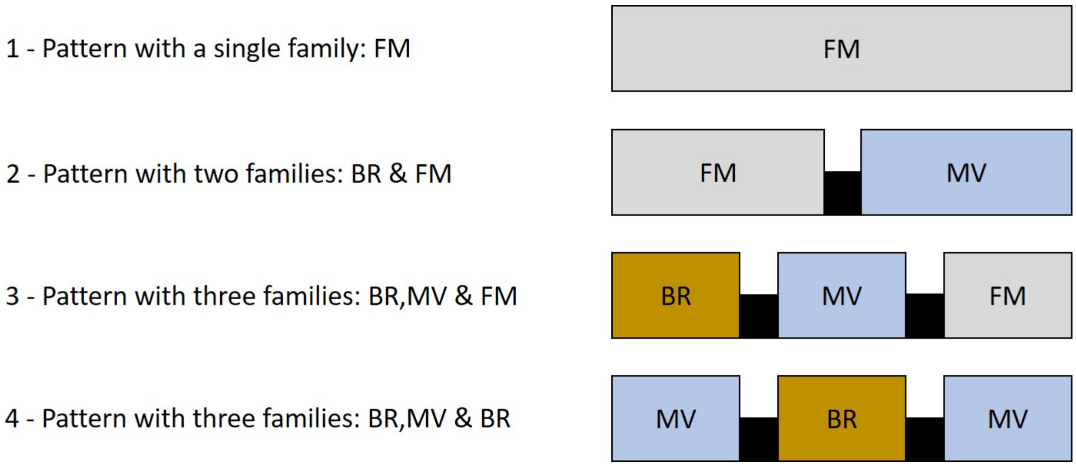

Figure 4.1. Sample illustrations for patterns including up to 3 families

Moreover, we distinguish the amount produced at the beginning, in the middle and at the end of a period for each family. Let us focus on Pattern 3 as an example.

Family BR corresponds to the beginning, MV to the middle and FM to the end. Note that as MV in Pattern 4, a family can also appear multiple times in different orders in a pattern. We assume that for patterns having at most two families, the set of families produced in the middle is empty.

In our formulations we will assign a pattern to each period. Consequently, it is also important that the setup between the last family of a predecessor pattern and the first family of its successor pattern is also feasible. Setup data is known and hence is an input. We can efficiently represent this data as a matrix having families in columns and rows. Each cell in the matrix corresponds to the setup duration/cost between the corresponding couple. Notice that, for infeasible family couples, which can be due to some technical properties, cells can be filled up with a sufficiently large value being larger than maximum number of days in a month.

Let us explain our approach regarding the representation of the setup over period boundaries in more detail with the help of illustrations as in Figure 4.2. Case (a) is an example where the setup time spent between families MV and BR crosses over period boundary. The Case (b) represents an example where the setup time is spent at the beginning of successor period. Note that depending on the production quantity and consequently duration decisions, it might well be also spent at the end of the predecessor period as in case (c). Finally, case (d) is an example for no-setup instance as the production within the same family continues. Note that with this approach the model can decide on allocating patterns such that setup is executed during period boundaries, which is not possible with sequence decisions in [48].

4.1.2. Generation and Pre-processing

As explained in Section 4.1.1, a pattern is simply an ordered list of families that we can assign to a period on the production line. We can generate patterns with the algorithm shown in Figure 4.3.

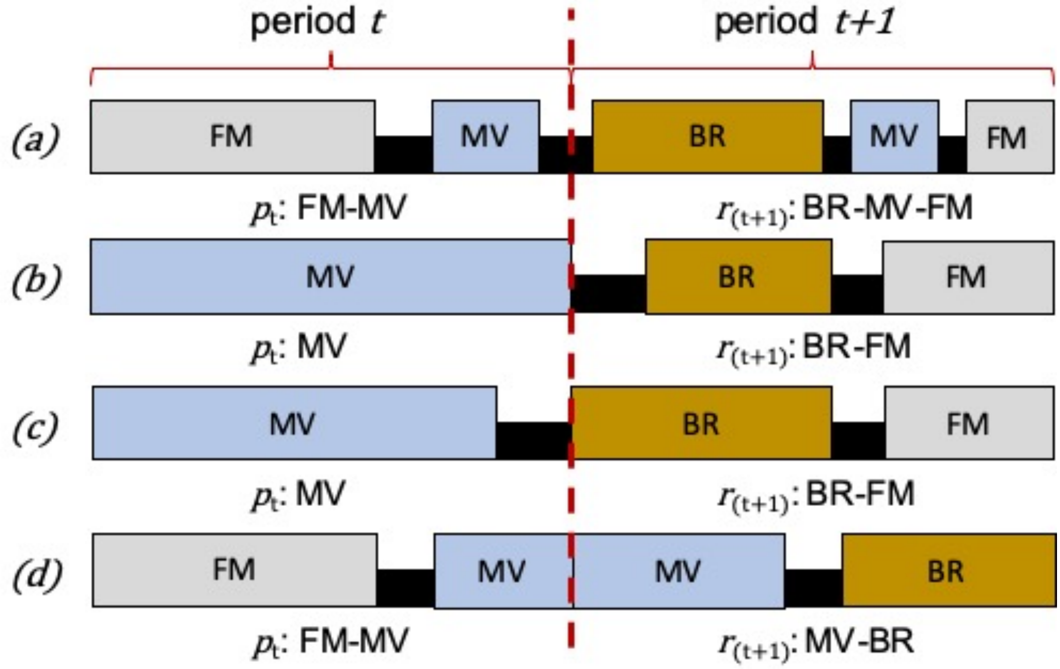

Figure 4.2. Sample setup illustrations

The algorithm works with the set of families F and the corresponding setup matrix M, which we use as input to a recursive procedure called *Extend*. At each call to *Extend*, the procedure evaluates each family f with respect to three criteria: i) f should be different than the last family of the current sequence, ii) by inserting f to the end of the sequence, minimum possible duration of this new sequence should not exceed the duration of a period, iii) if by adding f to the end of the sequence minimum possible duration exceeds the duration of a period, then there should be at least a strictly positive amount of time for producing f in addition to minimum possible duration of the sequence.

We define the minimum possible duration of a sequence as the sum of minimum production duration of appearing families and the setup required for the sequence. Also note that, with criteria iii), we make sure that even if a sequence is not feasible to be executed in a period with respect to its minimum duration, we do not eliminate it since our formulations can handle it. We explain this further in Sections 4.2.1 and 4.3.1 in detail. Note that, the algorithm generates all possible sequencing combinations.

GeneratePatterns (F, M)

inputs : Set F of all families and setup matrix M

$LL \leftarrow \emptyset$ (LL is a list)

return Extend(LL, F, M)

Extend (LL, F, M)

inputs : A list to be extended with new family insertions, set of families and setup matrix

$P \leftarrow \emptyset$

foreach *family* $f \in F$ **do**

  if $tail(LL) \neq f$ and CanAdd(LL, f, M) **then**

    $LL \leftarrow LL \cup f$

    $P \leftarrow P \cup LL$

    $P \leftarrow P \cup$ Extend(LL, F, M)

return P

CanAdd (LL, f, M)

inputs : A list and a family f and setup matrix

$D \leftarrow$MinDuration(LL, M)

if $D \geq$ length of a period **then**

  return $FALSE$

$S \leftarrow M[tail(LL), f]$

$D \leftarrow D + S$

if $D \geq$ length of a period **then**

  return $FALSE$

else

  return $TRUE$

MinDuration (LL, M)

inputs : A list and setup matrix

$D \leftarrow 0$

foreach *family* $f \in LL$ **do**

  $D \leftarrow D + M[prev(f), f] + MD_f$

return D

Figure 4.3. Generate all patterns p for a given set of families F.

We observe that multiple patterns generated with the algorithm in Figure 4.3 can result in the production of the same set of families for a given beginning and ending family pair. Let us elaborate with illustrative examples. Let f_1, f_2, f_3 and f_4 be a set of families and p_1 and p_2 be a couple of generated patterns containing these families. Let the sequence of p_1 be f_1 - f_2 - f_3 - f_4 and the sequence of p_2 be f_1 - f_3 - f_2 - f_4. If setup costs for pattern p_1 is less than that of p_2, then an optimal solution will favor p_1 to p_2 since both patterns have common starting and ending families, and the same set of families produced in only different sequences.

A similar redundancy appears in cases where a pattern contains as sub-sequence, the replication of a specific number of times of another pattern. Let f_1 and f_2 be a couple of families and p_1 and p_2 be a couple of generated patterns. Let the sequence of p_1 be f_1 - f_2 and the sequence of p_2 be f_1 - f_2 - f_1 - f_2. Notice that p_1 is a *'shrunk'* version of p_2, and that since p_2 yields more setup time and setup cost having twice the setup f_1 to f_2 and one f_2 to f_1 whereas p_1 yields more useful production time, p_2 can be removed from the list of patterns, thus reducing the number of binary variables in both formulations.

Algorithm in Figure 4.4 groups all patterns with respect to their canonical representation and keeps the one having the least associated cost from each group. Since we need to keep all the patterns enabling all possible transitions over period boundaries, information about the beginning and the ending families should not be lost, which we ensure by sub procedure *GetCanonicalRepresentation* in Figure 4.4.

4.2. Pattern Transition Based Model

Table 4.2 lists the symbols used in PTBM in addition to common symbols listed in Table 4.1 We present the constraints in Section 4.2.1. First, we define the fundamental constraints of GLSP followed by the constraints related to business model, which are tied to specifics of float glass manufacturing. Finally, we present the campaign defining constraints. We define the objective function and complete model in Section 4.2.2.

Table 4.2. Symbols used in PTBM.

Parameter	Description
STP_f	Maximum setup time in days such that product family f is predecessor
STS_f	Maximum setup time in days such that product family f is successor
MST_{fg}	Maximum setup time in days needed for switching from product family f to any other family or from any family to family g, can similarly be expressed as $max(STP_f, STS_g)$
f_p^H	First family in pattern p
f_p^T	Last family in pattern p
Variable	**Description**
θ_{prt}	Auxiliary variable indicating whether machine switched from pattern p to pattern r at the beginning of period t
F_{pt}	Setup time, in days, spent for pattern p at the beginning of period t
B_{pt}	Setup time, in days, spent for pattern p at the end of period t

SimplifyPatterns (P)

inputs : Set of patterns P

$P' \leftarrow \emptyset$

$G \leftarrow$ Group all patterns in P in by GetCanonicalRepresentation(p)

foreach pattern group $g \in G$ **do**

  $P' \leftarrow P' \cup \text{argmin}_p = \{c_p\}$

return P'

GetCanonicalRepresentation (p)

inputs : A pattern p

$f \leftarrow$ beginning family of pattern p

$g \leftarrow$ ending family of pattern p

$M \leftarrow$ ordered distinct list of families in pattern p

$s \leftarrow concatenate(f, f' \in M, g)$

return s

Figure 4.4. Pattern preprocessing algorithm.

In order to facilitate the understanding of the formulation logic, we present Figure 4.5 as an illustrative example. We have patterns *FM-MV* and *BR-MV-FM* assigned to periods t and $t+1$ respectively, and the relations between periods in terms of variables are available on the figure. Moreover, considering pattern BR-MV-FM assigned to period $t+1$, let us note that family BR is produced in order b at the beginning, MV in m in the middle and FM in e at the end.

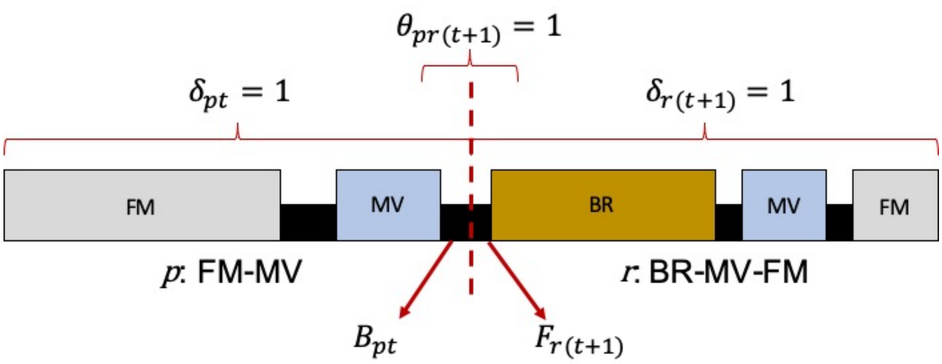

Figure 4.5. Illustration of PTBM decisions

4.2.1. Constraints

We permit backlog for demand satisfaction since the demands of products can be spread over the planning horizon whereas the duration and the timing of production campaigns are restricted. Eq. (4.1) ensures the consistency of demand satisfactions.

$$\sum_{\substack{k \in T \\ k \geq t}} S_{jtk} + U_{jt} = D_{jt} \quad \forall\, j \in J,\ t \in T \tag{4.1}$$

Eq. (4.2) is the inventory balance constraint that links production quantity X, ending inventory I and demand satisfaction S variables across time periods.

$$I_{j(t-1)} + X_{jt} - \sum_{\substack{k \in T \\ k \leq t}} S_{jkt} = I_{jt} \quad \forall\, j \in J,\ t \in T \tag{4.2}$$

Production cannot be interrupted since the furnace needs to be up and running in 24/7 operating mode. Available capacity must hence be fully utilized, which is ensured by Eq. (4.3). Note that in addition to time spent for production, Eq. (4.3) incorporates the setup time required due to the pattern selection.

$$\sum_{o \in O} d^o_{ft} + \sum_{p \in P} (ST_p \delta_{pt} + F_{pt} + B_{pt}) = A_t \quad \forall\, t \in T \tag{4.3}$$

We define the auxiliary variables d^o corresponding to the number of days allocated for production of family f at the beginning, in the middle or at the end of a period t. We relate d^o to the production quantity variables X with Eq. (4.4).

$$\sum_{j \in J} v_j X_{jt} = \sum_{o \in O} d^o_{ft} \quad \forall\, f \in F,\ t \in T \tag{4.4}$$

Due to the physical and the chemical nature of float glass production, random errors are observed on glass surface. Moreover, products can be substituted. This is with respect to their size group and quality attributes, namely s and q.

For example, a glass sheet of size group s can be cut into smaller sizes. Similarly, a sheet of quality q can be substituted as an item of lower quality. Furthermore, depending on the characteristics of the production line, production amount of a specific size group s and quality q cannot exceed a certain percentage of the total production quantity within a time period. Consequently, various production compositions are feasible. We denote this phenomenon as partially controllable co-production as explained in Chapter 3. Eq. (4.5) ensures that the production quantities in a time period yield a feasible composition within a specific family. The rates R_{fqs} depend on the characteristics of each furnace and are driven from the historical production data. Note that this approach is defined in [2].

$$\sum_{\substack{j \in J(f) \\ Q(j) \leq q \\ S(j) \leq s}} X_{jt} \leq \sum_{j \in J(f)} X_{jt}\ R_{fqs} \quad \forall\ f \in F,\ q \in Q,\ s \in S,\ t \in T \tag{4.5}$$

Our approach for the campaign planning is mainly based on assigning patterns to time periods. Eq. (4.6) ensures that a single pattern is assigned to each period.

$$\sum_{p \in P} \delta_{pt} = 1 \quad \forall\ t \in T \tag{4.6}$$

To ensure the efficiency and the stability of the manufacturing process, a minimum production duration should be ensured for each run of a product family. Eq. (4.7) models this requirement, ensuring a lower bound for production duration of families that are produced in the middle of a pattern. Considering the period boundaries, in an optimum solution the minimum duration can be split into two adjacent periods. Hence, we introduce Eq. (4.8). On the other hand, we need to set proper upper bounds on the production duration variables. Eq. (4.9) ensures that producing family f in order o is permitted only if a corresponding pattern is assigned in that period.

$$d^m_{ft} \geq MD_{fp}\delta_{pt} \quad \forall\ p \in P,\ f \in F^m(p),\ t \in T \tag{4.7}$$

$$d^e_{f(t-1)} + d^b_{ft} \geq MD_f\delta_{pt} \quad \forall\ p \in P,\ f \in F^b(p),\ t \in T,\ t \geq 1 \tag{4.8}$$

$$d^o_{ft} \leq \sum_{p \in P^o(f)} A_t \delta_{pt} \quad \forall\ f \in F,\ o \in O,\ t \in T \tag{4.9}$$

In order to properly handle setup crossover, we need to relate θ variables with δ variables. This can be formulated as in Eq. 4.10, which is a non-linear constraint.

$$\theta_{prt} = \delta_{p(t-1)} \delta_{rt} \quad \forall\ p, r \in P,\ t \in T,\ t \geq 1 \tag{4.10}$$

Note that we can linearize Eq. 4.10 as in Eqs. (4.11)–(4.13). Hence, we do not consider Eq. (4.10) any further. Moreover, Eqs. (4.11)–(4.13) permit relaxation of θ variables as $\theta_{prt} \geq 0$

$$\theta_{prt} \leq \delta_{p(t-1)} \quad \forall\ p, r \in P,\ t \in T,\ t \geq 1 \tag{4.11}$$

$$\theta_{prt} \leq \delta_{rt} \quad \forall\ p, r \in P,\ t \in T \tag{4.12}$$

$$\theta_{prt} \geq \delta_{p(t-1)} + \delta_{rt} - 1 \quad \forall\ p, r \in P,\ t \in T,\ t \geq 1 \tag{4.13}$$

Setup time spent at the beginning and at the end of a period t are managed with Eqs. (4.14)–(4.15). Note that these are big-M type constraints with MST_{fg} being the tightest big-M value. When a pattern transition is active through θ variable, setup time for the corresponding family pair is binding for the sum of setup time variables B and F. Otherwise, both upper bound and lower bound become redundant. Notice that it may or may not be the case that the setup time spans period boundaries with our approach.

$$ST_{fg} + MST_{fg}(1 - \theta_{prt}) \geq B_{p(t-1)} + F_{rt} \quad \forall\ p, r \in P,$$
$$f = f^T_p, g = f^H_R,\ t \in T,\ t \geq 1 \tag{4.14}$$

$$ST_{fg} - MST_{fg}(1 - \theta_{prt}) \leq B_{p(t-1)} + F_{rt} \quad \forall\ p, r \in P,$$
$$f = f^T_p, g = f^H_R,\ t \in T,\ t \geq 1 \tag{4.15}$$

It is also imperative that the variables for setup time at the beginning and at the ending of a period are zero unless the corresponding pattern is selected. Eqs. (4.16)–(4.17) ensure this requirement.

$$F_{pt} \leq STS_f \delta_{pt} \quad \forall\, p \in P, f = f_p^H,\ t \in T,\ t \geq 1 \tag{4.16}$$

$$B_{pt} \leq STP_f \delta_{pt} \quad \forall\, p \in P, f = f_p^T,\ t \in T,\ t \geq 0 \tag{4.17}$$

It might be the case that, switching from a certain product family f to another g is not possible due to some technical restrictions or business practice. Eq. (4.18) ensures that the model does not generate such an output.

$$\delta_{p(t-1)} + \delta_{rt} \leq 1 \quad \forall\, p, r \in P, f = f_p^T, g = f_r^H,\ (f, g) \in \Gamma(f, g),\ t \in T,\ t \geq 1 \tag{4.18}$$

4.2.2. Objective and Complete Model

We define the objective function as cost minimization. We assume that production cost for each product j remains constant within the planning horizon. Inventory holding costs for each product is driven from its production cost. Hence, production costs are implicitly included in the model and do not appear in the objective. We sum inventory holding and demand satisfaction costs over products and periods as the first three components. Our approach for demand unsatisfaction is based on the assumption that it is favorable to satisfy a demand, no matter how long the backlog period is, over unsatisfying. To achieve this, the cost associated with unsatisfaction is calculated as $b_j\ (|T| - t + 1)$, which reflects our assumption that demand can be satisfied from an infinite capacity after the planning horizon ends with a corresponding backlog cost associated. In addition, having the coefficient set as $(|T| - t + 1)$ earlier demands will be satisfied more preferably. Moreover, the cost associated to each family setup is significant and we incorporate this cost into the objective function with both pattern selection and pattern transition variables as with last two components. Model 4.1 represents the complete formulation for PTBM.

Model 4.1. *Pattern Transition Based Model (PTBM)*

$$\begin{aligned} \textit{Minimize} \sum_{\substack{j\in J\\ t\in T}} & \left[h_j \; I_{jt} + b_j \; (|T| - t + 1) \; U_{jt} + \sum_{\substack{k\in T\\ k\le t}} (b_j \; (t-k) \; S_{jkt}) \right] \\ & + \sum_{p\in P} c_p \delta_p + \sum_{\substack{t\in T\\ f=f_p^T\\ g=f_r^H\\ (f,g)\notin\Gamma(f,g)}} c_{fg}\theta_{prt} \\ \textit{subject to} \; & (4.1)\text{–}(4.9) \\ & (4.11)\text{–}(4.18) \\ & I_{jt}, X_{jt}, U_{jt} \ge 0 \quad \forall (j\in J, t) \\ & S_{jtk} \ge 0 \quad \forall (j \in J, t, k \ge t) \\ & \delta_{pt} \in \{0,1\} \quad \forall (p,t) \\ & F_{pt}, B_{pt}, \theta_{pt} \ge 0 \quad \forall (p,t) \end{aligned}$$

4.3. Family Transition Based Model

In PTBM, an auxiliary variable θ_{prt} is introduced for each feasible pattern pair and time period. This approach may be inefficient in cases where there are multiple pattern couples such that the predecessor's last family and the successor's first family are same. This leads us to the main idea in FTBM. The main difference in FTBM is the way we formulate the transition between periods. Instead of introducing an auxiliary variable for each feasible pattern couple, we introduce variables for a distinct set of family pairs corresponding to one or more pattern pair transition.

Table 4.3 lists the symbols used in FTBM in addition to the common symbols listed in Table 4.1. We present the constraints in Section 4.3.1. Figure 4.6 illustrates the formulation logic and the variable mapping to a possible campaign plan. Notice that the campaign plan is the same as the one illustrated for PTBM in Figure 4.5.

Table 4.3. Symbols used in FTBM.

Set	Description
$P^S(f)$	Subset of patterns whose first family is f
$P^E(f)$	Subset of patterns whose last family is f
Variable	**Description**
γ^S_{ft}	Indicator for selection of family f as starting in period t
γ^E_{ft}	Indicator for selection of family f as ending in period t
θ_{fgt}	Auxiliary variable indicating whether machine switched from family f to family g at the beginning of period t
n^P_{fgt}	Number of days spent for setup in predecessor period $t-1$ for switched from family f to family g at the beginning of period t
n^S_{fgt}	Number of days spent for setup in successor period t for switched from family f to family g at the beginning of period t
F_t	Setup time, in days, spent at the beginning of period t
B_t	Setup time, in days, spent at the end of period t

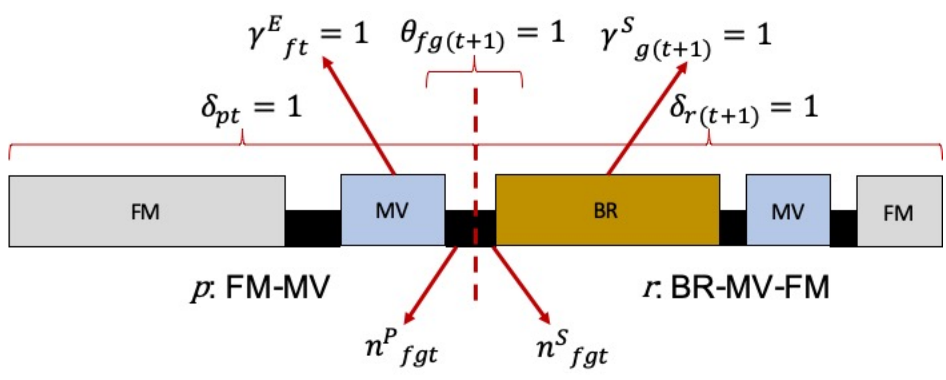

Figure 4.6. Illustration of FTBM decisions

4.3.1. Constraints

First, we note that since FTBM differs from PTBM with respect to the formulation of the state transition over period boundaries, some other concepts remain the same. Hence, the corresponding constraints are still valid for FTBM. In particular, requirement and inventory balance constraints with Eqs. (4.1)–(4.2), Eq. (4.4), which relates production duration and quantity variables, and Eq. (4.5) formulating the production composition regarding the size group and the quality are included in FTBM. Similarly, Eq. (4.6) ensuring assignment of a single pattern in each period and Eqs. (4.7)–(4.9) ensuring the minimum duration for producing family f are also valid for FTBM. Resource balance constraints, that are defined with Eq. (4.3) in Section 4.2.1 need to be modified due to the differences in the definitions of setup related variables F and B. Note that they do not depend on pattern p in FTBM but rather only on period t. Eq. (4.19) formulates resource balance as follows:

$$\sum_{o \in O} d^o_{ft} + \sum_{p \in P} ST_p \delta_{pt} + F_t + B_t = A_t \quad \forall\, t \in T \tag{4.19}$$

In order to determine the first and the last family produced in a period we set Eqs. (4.20)–(4.21). Notice that with Eq. (4.6) combined with Eqs. (4.20)–(4.21), variables (γ^S, γ^E) can only have values from $\{0, 1\}$. Hence, we can relax them as $\gamma^S, \gamma^E \geq 0$.

$$\gamma^S_{ft} = \sum_{p \in P^S(f)} \delta_{pt} \quad \forall\, f \in F,\ t \in T \tag{4.20}$$

$$\gamma^E_{ft} = \sum_{p \in P^E(f)} \delta_{pt} \quad \forall\, f \in F,\ t \in T \tag{4.21}$$

θ variables indicate whether a changeover is performed from family f to family g at the beginning of period t, and hence are binary. Similar to Eq. (4.10), θ variables should be equal to 1 if and only if both corresponding γ variables are equal to 1.

Eqs. (4.22)–(4.24) allow us to linearize and relax θ as $\theta \geq 0$.

$$\theta_{fgt} \leq \gamma^E_{f(t-1)} \quad \forall\, f, g \in F,\ t \in T,\ t \geq 1 \tag{4.22}$$

$$\theta_{fgt} \leq \gamma^S_{gt} \quad \forall\, f, g \in F,\ t \in T \tag{4.23}$$

$$\theta_{fgt} \geq \gamma^E_{f(t-1)} + \gamma^S_{gt} - 1 \quad \forall\, f, g \in F,\ t \in T,\ t \geq 1 \tag{4.24}$$

Eq. (4.25) ensures that necessary setup time is allocated for color transition.

$$n^P_{fgt} + n^S_{fgt} = ST_{fg}\theta_{fgt} \quad \forall\, f, g \in F,\ (f,g) \notin \Gamma(f,g),\ t \in T \tag{4.25}$$

We relate setup time variables for families (n^S, n^E) to period based variables (F, B) with Eqs. (4.26)–(4.27).

$$F_t = \sum_{(f,g) \notin \Gamma(f,g)} n^S_{fgt} \quad \forall\, t \in T \tag{4.26}$$

$$B_t = \sum_{(f,g) \notin \Gamma(f,g)} n^P_{fg(t+1)} \quad \forall\, t \in T \tag{4.27}$$

Eq. (4.28) ensures that no infeasible family transition is permitted. Note that this is the counterpart of Eq. (4.18).

$$\gamma^E_{f(t-1)} + \gamma^S_{gt} \leq 1 \quad \forall\, f, g \in F,\ (f,g) \in \Gamma(f,g),\ t \in T,\ t \geq 1 \tag{4.28}$$

4.3.2. Objective and Complete Model

The objective function is the same as PTBM. Model 4.2 represents the complete formulation for FTBM.

Model 4.2. *Family Transition Based Model (FTBM)*

$$
\begin{aligned}
\textit{Minimize} \ & \sum_{\substack{j \in J \\ t \in T}} \left[h_j \ I_{jt} + b_j \ (|T| - t + 1) \ U_{jt} + \sum_{\substack{k \in T \\ k \leq t}} (b_j \ (t-k) \ S_{jkt}) \right] \\
& + \sum_{p \in P} c_p \delta_p + \sum_{\substack{t \in T \\ (f,g) \notin \Gamma(f,g)}} c_{fg} \theta_{fgt} \\
\textit{subject to} \ & (4.1)\text{–}(4.2) \\
& (4.4)\text{–}(4.9) \\
& (4.19)\text{–}(4.28) \\
& I_{jt}, X_{jt}, U_{jt} \geq 0 \quad \forall (j \in J, c, t) \\
& S_{jtk} \geq 0 \quad \forall (j \in J, t, k \geq t) \\
& \delta_{pt} \in \{0, 1\} \quad \forall (p, t) \\
& \theta_{fgt} \geq 1 \quad \forall (f, g, t) \\
& \gamma^S_{ft}, \gamma^E_{ft} \geq 0 \quad \forall (f, t) \\
& F_t, B_t \geq 0 \quad \forall (t) \\
& n^P_{fgt}, n^S_{fgt} \geq 0 \quad \forall (f, g, t)
\end{aligned}
$$

4.4. Comparison of Pattern and Family Transition Based Models

As explained in detail in Sections 4.2 and 4.3, formulations differ from each other with respect to the formulation of the state transition over period boundaries. In PTBM, there is a θ variable for each pair of patterns whereas in FTBM θ variables are mapped to each pair of families. The FTBM associates state decision variables δ to setup duration through a convex hull reformulation with Eqs. (4.20), (4.21) and (4.25). Hence, we argue that FTBM is tighter than PTBM with the following proposition.

Proposition 4.1. *Let S^{FTBM} and S^{PTBM} be the feasible regions of linear programming relaxations of FTBM and PTBM respectively. Then, $S^{FTBM} \subset S^{PTBm}$.*

Proof. Let I be the set of family pairs (f', g') such that $\theta_{f'g't} > 0$ in a feasible solution to PTBMV. Then summing Eq. (4.25) over $(f', g') \in I$, we obtain

$$\sum_{(f',g')\in I} n^P_{f'g't} + \sum_{(f',g')\in I} n^P_{f'g't} = \sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \tag{4.29}$$

Note that, the first term is equal to F_{t+1} and the second term is equal to B_t on the left hand side of the equation. Moreover, from Eqs. (4.22)–(4.24), we obtain following inequalities respectively by again summing over $(f', g') \in I$.

$$\sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \leq \sum_{(f',g')\in I} ST_{f'g'}\gamma^E_{f't} \tag{4.30}$$

$$\sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \leq \sum_{(f',g')\in I} ST_{f'g'}\gamma^S_{g'(t+1)} \tag{4.31}$$

$$\sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \geq \sum_{(f',g')\in I} (\gamma^E_{f't} + \gamma^S_{g'(t+1)}) + |I| \tag{4.32}$$

Left hand side of all these three inequalities can hence be replaced by $F_{t+1} + B_t$. On the other hand, when we sum Eqs. (4.16) and (4.17) followed by another sum over $(p', r') \in J$ where p' and r' correspond to patterns having f' as ending family and g' as starting family respectively, we obtain

$$\sum_{(p',r')\in J} (B_{p't} + F_{r'(t+1)}) \leq \sum_{(p',r')\in J} (STP_{f'}\delta_{p't} + STS_{g'}\delta_{r'(t+1)}) \tag{4.33}$$

which also has the left hand side equal to $F_{t+1}+B_t$. Summing Eq. (4.14) over $(p', r') \in J$ gives

$$\sum_{(p',r')\in J} ST_{f'g'} - \sum_{(p',r')\in J} MST_{f'g'} + \sum_{(p',r')\in J} \theta_{p'r't} \leq \sum_{(p',r')\in J} (B_{p't} + F_{r'(t+1)}) \tag{4.34}$$

Note that right hand side of the inequality (4.34) is also equal to $F_{t+1} + B_t$.

Then from Eq. (4.30) and Eq. (4.31), we obtain following inequalities which are always true by definition of $STP_{f'}$ and $STS_{g'}$ with respect to $ST_{f'g'}$

$$\sum_{(f',g')\in I} ST_{f'g'}\gamma^E_{f't} \leq \sum_{(p',r')\in J} (STP_{f'}\delta_{p't} + STS_{g'}\delta_{r't}) \tag{4.35}$$

$$\sum_{(f',g')\in I} ST_{f'g'}\gamma^S_{g'(t+1)} \leq \sum_{(p',r')\in J} (STP_{f'}\delta_{p't} + STS_{g'}\delta_{r't}) \tag{4.36}$$

Finally from Eq. (4.34) we obtain

$$\sum_{(p',r')\in J} ST_{f'g'} - \sum_{(p',r')\in J} MST_{f'g'} + \sum_{(p',r')\in J} \theta_{p'r'(t+1)} \leq \sum_{(f',g')\in I} (\gamma^E_{f't} + \gamma^S_{g'(t+1)}) - |I| \tag{4.37}$$

The first to components of the left hand side is negative by definition of $ST_{f'g'}$ and $MST_{f'g'}$. Exploring the third component from Eq. (4.13) by summing over $(p',r')\in J$

$$\sum_{(p',r')\in J} \delta_{p't} + \sum_{(p',r')\in J} \delta_{r'(t+1)} - |J| \leq \sum_{(p',r')\in J} \theta_{p'r't} \tag{4.38}$$

Since, $\sum_{(p',r')\in J} \delta_{p't} = \sum_{(f',g')\in I} \gamma^E_{f't}$, $\sum_{(p',r')\in J} \delta_{r'(t+1)} = \sum_{(f',g')\in I} \gamma^S_{g'(t+1}$ and $|J| \geq |I|$, then (4.37) is also always true. Hence, for each fractional solution to S^{FTBM}, one can find a corresponding solution in S^{PTBM}.

On the other hand, let p^{FM1} and p^{FM2} be two patterns ending with family FM and allocated have corresponding δ variables equal to 0.5 and 0.5 in period t respectively in a feasible solution to PTBM. Similarly, let r^{FM3} and r^{MV4} be two patterns starting with families FM and MV respectively with corresponding δ variables equal to 0.4 and 0.6 in period $t+1$. Following Eqs. (4.11)–(4.13) variable $\theta_{p^{(FM3)(t+1)}r^{(MV4)(t+1)}} \geq 0 \geq (0.5+0.4-1)$. Then Eq. (4.14) and Eq. (4.15), will become redundant since θ can take value of zero. However, in FTBM, the corresponding θ variable, namely $\theta_{(FM3)(MV4)(t+1)}$, has a lower bound of 0.6 from Eq. (4.24). This triggers Eq. (4.25) such that the left hand side has to equal $ST_{(FM3)(MV4)} * 0.6$.

This might results in different setup duration for PTBM and FTBM. Hence there exists a solution to the LP relaxation of PTBM, which is not a feasible solution of the LP relaxation of FTBM. □

4.5. Formulation Variations

In both formulations PTBM and FTBM, infeasible changeovers between families over period boundaries are prohibited explicitly with Eq. (4.18) and Eq. (4.28) in PTBM and FTBM, respectively. From another point of view, this is equivalent to the condition that over period boundaries, only feasible family setups should be allowed. Hence, this can be achieved with Eq. (4.39) for PTBM:

$$\sum_{\substack{p,r \in P \\ f = f_p^T \\ g = f_r^H \\ (f,g) \notin \Gamma(f,g)}} \theta_{prt} = 1 \quad \forall\, t \in T,\ t \geq 1 \tag{4.39}$$

and with Eq. (4.40) for FTBM:

$$\sum_{\substack{f,g \in F \\ (f,g) \notin \Gamma(f,g)}} \theta_{fgt} = 1 \quad \forall\, t \in T,\ t \geq 1 \tag{4.40}$$

Notice that Eqs. (4.39)–(4.40) may decrease the number of constraints significantly depending on the number of patterns and families. In PTBM and FTBM, Eq. (4.18) and Eq. (4.28) are written explicitly for each period transition and for each pair of infeasible pattern and family pairs respectively. On the other hand, in variant models PTBMV and FTBMV, a single equation exists as Eq. (4.39) and Eq. (4.40) for each period transition. Model 4.3 and Model 4.4 represent the complete formulation for PTBMV and FTBMV respectively.

Model 4.3. *Pattern Transition Based Model Variant (PTBMV)*

$$\begin{aligned}
\textit{Minimize} \sum_{\substack{j\in J\\ t\in T}} & \left[h_j \ I_{jt} + b_j \ (|T| - t + 1) \ U_{jt} + \sum_{\substack{k\in T\\ k\leq t}} (b_j \ (t-k) \ S_{jkt}) \right] \\
& + \sum_{p\in P} c_p \delta_p + \sum_{\substack{t\in T\\ (f,g)\notin\Gamma(f,g)}} c_{fg}\theta_{fgt}
\end{aligned}$$

subject to (4.1)–(4.9)

(4.11)–(4.17)

(4.39)

$I_{jt}, X_{jt}, U_{jt} \geq 0 \quad \forall(j \in J, t)$

$S_{jtk} \geq 0 \quad \forall(j \in J, t, k \geq t)$

$\delta_{pt} \in \{0, 1\} \quad \forall(p, t)$

$F_{pt}, B_{pt}, \theta_{pt} \geq 0 \quad \forall(p, t)$

Model 4.4. *Family Transition Based Model Variant (FTBMV)*

$$\begin{aligned}
\textit{Minimize} \sum_{\substack{j\in J\\ t\in T}} & \left[h_j \ I_{jt} + b_j \ (|T| - t + 1) \ U_{jt} + \sum_{\substack{k\in T\\ k\leq t}} (b_j \ (t-k) \ S_{jkt}) \right] \\
& + \sum_{p\in P} c_p \delta_p + \sum_{\substack{t\in T\\ (f,g)\notin\Gamma(f,g)}} c_{fg}\theta_{fgt}
\end{aligned}$$

subject to (4.1)–(4.2)

(4.4)–(4.9)

(4.19)–(4.27)

(4.40)

$I_{jt}, X_{jt}, U_{jt} \geq 0 \quad \forall(j \in J, t)$

$S_{jtk} \geq 0 \quad \forall(j \in J, t, k \geq t)$

$\delta_{pt} \in \{0, 1\} \quad \forall(p, t)$

$0 \leq \theta_{fgt} \leq 1, n^P_{fgt}, n^S_{fgt} \geq 0 \quad \forall(f, g, t)$

$$\gamma^S_{ft}, \gamma^E_{ft} \geq 0 \quad \forall (f,t)$$
$$F_t, B_t \geq 0 \quad \forall (t)$$

We argue that the variant formulations are tighter than primary formulations. The following proposition shows that PTBMV is tighter than PTBM.

Proposition 4.2. *Let S^{PTB} and S^{PTBV} be the feasible regions of linear programming relaxations of PTBM and PTBMV respectively. Then, $S^{PTBV} \subset S^{PTB}$.*

Proof. Let I be the set of pattern pairs (p', r') such that $\theta_{p'r'(t+1)} > 0$ in a feasible solution to PTBMV. Then, for each (p', r') we have

$$\delta_{p't} \geq \theta_{p'r'(t+1)}$$
$$\delta_{r'(t+1)} \geq \theta_{p'r'(t+1)}$$

from Eqs. (4.11)–(4.12) and since $\sum_{(p',r' \in I)} \theta_{p'r'(t+1)} = 1$ by Eq. (4.39), then we have

$$\sum_{(p',r') \in I} \delta_{p't} = \sum_{(p',r') \in I} \delta_{r'(t+1)} = 1$$

Hence,

$$\sum_{(p'',r'') \notin I} \delta_{p''t} = \sum_{(p'',r'') \notin I} \delta_{r''(t+1)} = 0$$

Note that such pattern couples include both feasible and infeasible pattern pairs and such feasible pairs Eq. (4.18) is not relevant. Moreover, for pairs $(p', r') \in I$ such that (p', r') setup is infeasible, since $\sum_{(p',r' \in I)} \theta_{p'r'(t+1)} = 1$ by assumption, we have $\delta_{p't} + \delta_{r'(t+1)} \leq 1$. Hence, each fractional solution of PTBMV is also feasible with respect to PTBM.

On the other hand, let f_1, f_2, f_3, f_4, f_5 and f_6 be families with no feasible transition between any couple except within same family. Let us note patterns including single families also as f_1, f_2 etc. Let δ values in a solution of PTBM be $\delta_{f_1 t} = 0.4$, $\delta_{f_2 t} = 0.5$, $\delta_{f_3 t} = 0.1$, $\delta_{f_1(t+1)} = 0.4$, $\delta_{f_4(t+1)} = 0.5$ and $\delta_{f_5(t+1)} = 0.1$. Note that since there is no feasible transition between any couples other than $f_1 t$ to $f_1(t+1)$, for any combination Eq. (4.18) is satisfied. However, since the only feasible transition ($f_1 t$ to $f_1(t+1)$) implies that $\theta_{f_1 f_1 (t+1)} \leq 0.4$ then Eq. (4.39) is violated and hence there exists a solution for the LP relaxation of PTBM, which is not feasible for the LP relaxation of PTBMV. $\square$

Note that by similar approach, we can also prove the following proposition.

Proposition 4.3. *Let S^{FTB} and S^{FTBV} be the feasible regions of linear programming relaxations of FTBM and FTBMV respectively. Then, $S^{FTBV} \subset S^{FTB}$.*

5. PARALLEL MACHINE PROBLEM

In this chapter, we extend the mathematical models for the campaign planning problem formulated in Chapter 4 to parallel machines. Main effects of parallel machines relate to production speed, size group and quality ratios, setup times and costs. Note that we extend the variant formulations and name the models Pattern Transition Based Model Variant on Parallel Machines (PTBMV-PM) and Family Transition Based Model Variant on Parallel Machines (FTBMV-PM) respectively. In addition, Table 5.1 illustrates symbols used in both PTBMV-PM and FTBMV-PM.

Table 5.1. Symbols used in both parallel machine formulations.

Set	**Description**
J	Set of products
R	Set of production lines
Q	Set of quality groups
S	Set of size groups
T	Set of time periods
P	Set of campaign patterns
F	Set of product families
O	Set of orders for timing of production in a period (b: beginnig, m: middle, e: end)
$P(f)$	Set of patterns containing family f at least once
$F(p)$	Set of families belonging to pattern p
$F^o(p)$	Set of families appearing in order o in pattern p
$J(f)$	Set of products belonging to family f
$\Gamma^r(f,g)$	Set of product family couples that are infeasible on production line r, $f, g \in F$

Table 4.1. Symbols used in both parallel machine formulations. (cont.)

Parameter	**Description**
D_{jt}	Demand of product j in period t
$I_{j(-1)}$	Beginning inventory of product j
v_{jr}	Production speed of product j on production line r
A_t	Available capacity of production lines in period t
$S(j)$	Index of the size group of product j
$Q(j)$	Index of the quality group of product j
R_{fqsr}	Maximum production ratio for quality group q and size group s for family f on production line r
MD_{fr}	Minimum production duration for family f on production line r
NT_{fp}	Number of times family f appears in the middle order of pattern p
ST_{pr}	Setup time needed for family order within pattern p on production line r
ST_{fgr}	Setup time needed for switching from product family f to family g on production line r
h_j	Inventory holding cost for product j
b_j	Cost of backlogging a demand of product j for a single period
u_{jr}	Unit production cost for producing product j on line r
c_{fgr}	Setup cost for switching from family f to family g on production line r
c_{pr}	Total setup cost for family order within pattern p on production line r
Variable	**Description**
I_{jt}	Inventory of product j at the end of period t
S_{jtk}	Satisfied quantity of demand from period t of product j in period k
U_{jt}	Unsatisfied quantity of demand from period t of product j
X_{jrt}	Production quantity of product j on production line r in period t

Table 4.1. Symbols used in both parallel machine formulations. (cont.)

Variable	Description
δ_{prt}	Binary indicator variable for selection of pattern p on production line r in period t
d^{o}_{frt}	Number of days spent for production of family f in order o on production line r in period t

5.1. Pattern Generation Extension

As explained in Section 4.1.1, a pattern is simply an ordered list of families to be assigned to a period of the single machine. Algorithm in Figure 4.3 generates patterns for a given set of families respecting setup feasibility and minimum production duration limitations. In parallel machine case, we can adapt this algorithm in a way that it runs for each machine. For each machine r, let F_r be the set of families that can be produced on r and M_r be the corresponding setup matrix. Hence, if we provide them to the *GeneratePatterns* procedure defined in Figure 4.3, then algorithm shown in Figure 5.1 will generate patterns for each r in set of machines, namely R

GenerateAllPatterns (R, F, M)
    inputs : Set R of machines, set F of all families and setup matrix M
    $LL \leftarrow \emptyset$ (LL is a list)

    foreach *machine* $r \in R$ **do**
        $F_r \leftarrow$ families that can be produced on machine r
        $M_r \leftarrow$ setup matrix associated with machine r
        $LL \leftarrow LL \cup$ GeneratePatterns(F_r, M_r)
    return LL

Figure 5.1. Generate all patterns p on all machines R.

5.2. Pattern Transition Based Model Variant on Parallel Machines

In this Section, we present the parallel machines extension of PTBMV, namely PTBMV-PM. Table 5.2 lists the symbols used in PTBMV-PM in addition to common symbols listed in Table 5.1 along with their brief descriptions. We present constraints in Section 5.2.1. We present the objective and formulate the complete model in Section 5.2.2.

Table 5.2. Symbols used in PTBMV-PM.

Parameter	**Description**
STP_{fr}	Maximum setup time such that product family f is predecessor on production line r
STS_{fr}	Maximum setup time such that product family f is successor on production line r
MST_{fgr}	Maximum setup time needed for switching from product family f to any other family or from any family to family g on production line r, can similarly be expressed as $max(STP_{fr}, STS_{gr})$
f_p^H	First family in pattern p
f_p^T	Last family in pattern p
Variable	**Description**
θ_{psrt}	Auxiliary variable indicating whether machine switched from pattern p to pattern s on production line r at the beginning of period t
F_{prt}	Setup time spent for pattern p on production line r at the beginning of period t
B_{prt}	Setup time spent for pattern p on production line r at the end of period t

5.2.1. Constraints

Eq. (5.1) ensures the consistency of demand satisfactions having demand backlog and unsatisfaction allowed.

$$\sum_{\substack{k \in T \\ k \geq t}} S_{jtk} + U_{jt} = D_{jt} \quad \forall\, j \in J,\ t \in T \tag{5.1}$$

Eq. (5.2) is the inventory balance constraint linking the production quantity X, inventory I and demand satisfaction S variables across time periods.

$$I_{j(t-1)} + \sum_{r \in R} X_{jrt} - \sum_{\substack{k \in T \\ k \leq t}} S_{jkt} = I_{jt} \quad \forall\, j \in J,\ t \in T \tag{5.2}$$

Eq. (5.3) formulates production line capacity usage ensuring 24/7 working. Note that since we define the capacity of production lines as time within a period, the capacity A_t does not depend on r.

$$\sum_{f \in F} \sum_{o \in O} d^{o}_{frt} + \sum_{p \in P} (ST_{pr}\delta_{prt} + F_{prt} + B_{prt}) = A_t \quad \forall\, r \in R,\ t \in T \tag{5.3}$$

Eq. (5.4) couples variables representing number of days of production allocated in an order for a family to production quantity variables. Order translates into the beginning, middle or ending of a period.

$$\sum_{j \in J} v_{jr} X_{jrt} = \sum_{o \in O} d^{o}_{frt} \quad \forall\, f \in F,\ r \in R,\ t \in T \tag{5.4}$$

Eq. (5.5) ensures that production quantities in a time period consist a feasible composition within a specific family on a production line.

$$\sum_{\substack{j \in J(f) \\ Q(j) \leq q \\ S(j) \leq s}} X_{jrt} \leq \sum_{j \in J(f)} X_{jrt}\ R_{fqrs} \quad \forall\, r \in R,\ f \in F,\ q \in Q,\ s \in S,\ t \in T \tag{5.5}$$

Eq. (5.6) ensures allocation of patterns to production lines for each period.

$$\sum_{p\in P}\delta_{prt}=1 \quad \forall\, r\in R,\ t\in T \tag{5.6}$$

Equations (5.7)–(5.9) serve to model a lower bound for production duration of families that are produced in the middle of a pattern, split into two adjacent periods and a proper upper bound, respectively.

$$d^m_{frt}\geq MD_{fr}NT_{fp}\delta_{pt} \quad \forall\, r\in R,\ p\in P,\ f\in F^m(p),\ t\in T \tag{5.7}$$

$$d^e_{fr(t)}+d^b_{fr(t+1)}\geq MD_{fr}\delta_{prt} \quad \forall\, r\in R,\ p\in P,\ f\in F^b(p)\cup F^e(p),\ t\in T \tag{5.8}$$

$$d^o_{frt}\leq\sum_{p\in P^o(f)}A_t\delta_{prt} \quad \forall\, r\in R,\ f\in F,\ o\in O,\ t\in T \tag{5.9}$$

Equations (5.10)–(5.12) serve for relating θ variables with δ variables to properly handle setup crossover.

$$\theta_{psrt}\leq\delta_{pr(t-1)} \quad \forall\, r\in R,\ p,s\in P,\ t\in T,\ t\geq 1 \tag{5.10}$$

$$\theta_{psrt}\leq\delta_{srt} \quad \forall\, r\in R,\ p,s\in P,\ t\in T \tag{5.11}$$

$$\theta_{psrt}\geq\delta_{pr(t-1)}+\delta_{rrt}-1 \quad \forall\, r\in R,\ p,s\in P,\ t\in T,\ t\geq 1 \tag{5.12}$$

Equations (5.13)–(5.14) manage setup time spent at the beginning and at the end of a period t.

$$ST_{fgr}+MST_{fgr}(1-\theta_{psrt})\geq B_{pr(t-1)}+F_{srt} \quad \forall\, r\in R,\ p,s\in P,\ f=f^T_p,g=f^H_s,\ t\in T,\ t\geq 1 \tag{5.13}$$

$$ST_{fgr}-MST_{fgr}(1-\theta_{psrt})\leq B_{pr(t-1)}+F_{srt} \quad \forall\, r\in R,\ p,s\in P,\ f=f^T_p,g=f^H_s,\ t\in T,\ t\geq 1 \tag{5.14}$$

Equations (5.15)–(5.16) ensure that setup time variables for period beginning and ending are zero unless the corresponding pattern is selected.

$$F_{prt} \leq STS_{fr}\delta_{prt} \quad \forall\, r \in R,\ p \in P, f = f_p^H,\ t \in T,\ t \geq 1 \tag{5.15}$$

$$B_{prt} \leq STP_{fr}\delta_{prt} \quad \forall\, r \in R,\ p \in P, f = f_p^T,\ t \in T,\ t \geq 0 \tag{5.16}$$

Model avoids infeasible family transitions with Eq. (5.17).

$$\sum_{\substack{p,s \in P \\ f = f_p^T \\ g = f_s^H \\ (f,g) \notin \Gamma^r(f,g)}} \theta_{psrt} = 1 \quad \forall\, r \in R,\ t \in T,\ t \geq 1 \tag{5.17}$$

5.2.2. Objective and Complete Model

We define the objective function as cost minimization. Other than having resource indices in δ and θ variables, the major difference is PTBMV-PM objective also includes production costs, namely u_{jr}. Demand satisfaction related costs are formulated with the same approach we explain in Section 4.2.2. Model 5.1 represents the complete formulation for PTBM.

Model 5.1. *Pattern Transition Based Model Variant on Parallel Machines (PTBMV-PM)*

$$\begin{aligned} \textit{Minimize} \ & \sum_{j \in J} \sum_{t \in T} \left[h_j \ I_{jt} + b_j \ (|T| - t + 1) \ U_{jt} + \sum_{\substack{k \in T \\ k \leq t}} (b_j \ (t - k) \ S_{jkt}) \right] \\ & + \sum_{r \in R} \sum_{j \in J} \sum_{t \in T} u_{jr} X_{jrt} + \sum_{r \in R} \sum_{p \in P} c_{pr} \delta_{pr} \\ & + \sum_{r \in R} \sum_{p \in P} \sum_{t \in T} \sum_{\substack{f = f_p^T \\ g = f_s^H \\ (f,g) \notin \Gamma^r(f,g)}} c_{fgr} \theta_{psrt} \end{aligned}$$

subject to (5.1)–(5.17)

$$I_{jt}, U_{jt} \geq 0 \quad \forall (j,t)$$

$$S_{jtk} \geq 0 \quad \forall (j,t,k \geq t)$$

$$d^{o}_{frt} \geq 0 \quad \forall (f,r,t)$$

$$X_{jrt} \geq 0 \quad \forall (j,r,t)$$

$$\delta_{prt} \in \{0,1\} \quad \forall (p,r,t)$$

$$F_{prt}, B_{prt}, \theta_{psrt} \geq 0 \quad \forall (p,s,r,t)$$

5.3. Family Transition Based Model Variant on Parallel Machines

In this Section, we present the parallel machine extension of FTBMV. Table 5.3 lists the symbols used in FTBMV-PM in addition to common symbols listed in Table 5.1 along with their brief descriptions. We present constraints in Section 5.3.1 and complete model with objective in Section 5.3.2.

5.3.1. Constraints

We defined requirement and inventory balance constraints with Equations (5.1)–(5.2) respectively. Since these concepts do not change in FTBM-PM, they're still valid. Similarly, Eq. (5.4) relating production duration and quantity variables and Eq. (5.5) formulating production composition regarding size group and quality are also valid for FTBM-PM. Lastly, Hence,(5.6) formulating pattern selection per production line in addition to minimum duration constraints formulated with Equations (5.7)–(5.9) are also valid in FTBMV-PM.

Eq. (5.18) is formulating resource balance as follows.

$$\sum_{f \in F} \sum_{o \in O} d^{o}_{frt} + \sum_{p \in P} ST_{pr} \delta_{prt} + F_{rt} + B_{rt} = A_t \quad \forall\, r \in R,\ t \in T \tag{5.18}$$

Table 5.3. Symbols used in FTBMV-PM.

Parameter	**Description**
$P^S(f)$	Patterns that family f is the first family
$P^E(f)$	Patterns that family f is the last family
Variable	**Description**
γ^S_{frt}	Indicator for selection of family f as starting on production line r in period t
γ^E_{frt}	Indicator for selection of family f as ending on production line r in period t
θ_{fgrt}	Auxiliary variable indicating whether production line r switched from family f to family g at the beginning of period t
n^P_{fgrt}	Number of days spent for setup on production line r in predecessor period $t-1$ for switched from family f to family g at the beginning of period t
n^S_{fgrt}	Number of days spent for setup on production line r in successor period t for switched from family f to family g at the beginning of period t
F_{rt}	Setup time spent on production line r at the beginning of period t
B_{rt}	Setup time spent on production line r at the end of period t

Equations (5.19)–(5.20) determine starting and ending family within a period. Notice that Eq. (5.6) combined with Equations (5.19)–(5.20), variables (γ^S, γ^E) can only have values from $\{0,1\}$ and hence be relaxed as $\gamma^S, \gamma^E \geq 0$.

$$\gamma^S_{frt} = \sum_{p \in P^S(f)} \delta_{prt} \quad \forall\, r \in R,\ f \in F,\ t \in T \tag{5.19}$$

$$\gamma^E_{frt} = \sum_{p \in P^E(f)} \delta_{prt} \quad \forall\, r \in R,\ f \in F,\ t \in T \tag{5.20}$$

θ variables indicate whether a changeover is performed from family f to family g at the beginning of period t on each production line, and they are related to γ variables with Equations (5.21)–(5.23).

$$\theta_{fgrt} \leq \gamma^E_{fr(t-1)} \quad \forall\, r \in R,\ f,g \in F,\ t \in T,\ t \geq 1 \tag{5.21}$$

$$\theta_{fgrt} \leq \gamma^S_{grt} \quad \forall\, r \in R,\ f,g \in F,\ t \in T \tag{5.22}$$

$$\theta_{fgrt} \geq \gamma^E_{fr(t-1)} + \gamma^S_{grt} - 1 \quad \forall\, r \in R,\ f,g \in F,\ t \in T,\ t \geq 1 \tag{5.23}$$

Eq. (5.24) ensures that each production line allocates necessary setup time for color transition.

$$n^P_{fgrt} + n^S_{fgrt} = ST_{fgr}\theta_{fgrt} \quad \forall\, r \in R,\ f,g \in F,\ (f,g) \notin \Gamma^r(f,g),\ t \in T \tag{5.24}$$

Equations (5.26)–(5.25) relate setup time variables for families (n^S, n^E) to period based variables (F, B).

$$F_{rt} = \sum_{(f,g) \notin \Gamma^r(f,g)} n^S_{fgrt} \quad \forall\, r \in R,\ f,g \in F,\ t \in T \tag{5.25}$$

$$B_{rt} = \sum_{(f,g) \notin \Gamma^r(f,g)} n^P_{fgr(t+1)} \quad \forall\, r \in R,\ f,g \in F,\ t \in T \tag{5.26}$$

Model avoids infeasible family transitions with Eq. (5.27).

$$\sum_{\substack{f,g \in F \\ (f,g) \notin \Gamma^r(f,g)}} \theta_{fgrt} = 1 \quad \forall\, r \in R,\ t \in T,\ t \geq 1 \tag{5.27}$$

5.3.2. Objective and Complete Model

The objective function is the same as PTBMV-PM explained in Section 5.2.2. Model 5.2 represents the complete formulation for PTBM.

Model 5.2. *Family Transition Based Model Variant with (FTBMV-PM)*

$$\begin{aligned}
\mathit{Minimize}\ & \sum_{j \in J} \sum_{t \in T} \left[h_j\ I_{jt} + b_j\ (|T| - t + 1)\ U_{jt} + \sum_{\substack{k \in T \\ k \leq t}} (b_j\ (t-k)\ S_{jkt}) \right] \\
& + \sum_{r \in R} \sum_{j \in J} \sum_{t \in T} u_{jr} X_{jrt} + \sum_{r \in R} \sum_{p \in P} c_{pr} \delta_{pr} \\
& + \sum_{r \in R} \sum_{p \in P} \sum_{t \in T} \sum_{\substack{f = f_p^T \\ g = f_s^H \\ (f,g) \notin \Gamma^r(f,g)}} c_{fgr} \theta_{psrt}
\end{aligned}$$

$$\begin{aligned}
\mathit{subject\ to}\ & (5.1)\text{–}(5.2) \\
& (5.4)\text{–}(5.9) \\
& (5.18)\text{–}(5.26) \\
& (5.27) \\
& I_{jt}, U_{jt} \geq 0 \quad \forall (j,t) \\
& S_{jtk} \geq 0 \quad \forall (j,t,k \geq t) \\
& d^o_{frt} \geq 0 \quad \forall f, r, t \\
& X_{jrt} \geq 0 \quad \forall (j,r,t) \\
& \delta_{prt} \in \{0,1\} \quad \forall (p,r,t) \\
& 0 \leq \theta_{fgrt} \leq 1 \quad \forall (f,g,r,t)
\end{aligned}$$

$$\gamma^S_{frt}, \gamma^E_{frt} \geq 0 \quad \forall (f, r, t)$$

$$F_{rt}, B_{rt} \geq 0 \quad \forall (r, t)$$

$$n^P_{fgrt}, n^S_{fgrt} \geq 0 \quad \forall (f, g, r, t)$$

5.4. Branch-and-Price Algorithm

In this Section, we introduce a new representation, namely *extended patterns*, for campaign plans enabling us formulate the problem in a simpler form, which drives the implementation of a branch-and-price (B&P) algorithm. We first describe the concept of extended pattern and present the reformulation based on the extended patterns. We then describe the CG specifics in Section 5.4.3, followed by the modeling of pricing problem as a shortest path problem in Section 5.4.4. Finally, we explain our initial column set generation, branching and node selection strategies, root node processing approach in addition to upper bound generation in Section 5.4.5.

5.4.1. Concept of Extended Pattern

Formulations FTBMV-PM and PTBMV-PM allocate a pattern from a pre-defined set of patterns, which are feasible in terms of minimum production duration of families involved, to each period of the production line. Moreover, a campaign plan is the sequence of families to be produced on a specific production line with start and end times of setups and production runs of families. A campaign plan is itself, from another point of view, another pattern covering the entire planning horizon. The reason is that is also a sequence of families to be executed on the corresponding line. Hence, we define each campaign plan as an *extended* pattern.

We define micro period as a unit amount of time multiplies of which can represent the continuous data we need to incorporate into our models. Recall from Section 3 that data with continuous time resolution includes setup times and minimum production duration of families.

Extended patterns represent the sequence of families and their respective durations, with the exception that durations are expressed as "number of micro periods". From another perspective, we divide the planning horizon into micro periods, and an *extended* pattern is an ordered representation of family allocations to each one of these micro periods. Figure 5.2 illustrates four different extended pattern examples for a set of two families F_1 and F_2. Families have 2 micro periods of minimum production duration each, and 1 and 3 micro periods of sequence-dependent setup times from F_1 to F_2 and from F_2 to F_1 respectively.

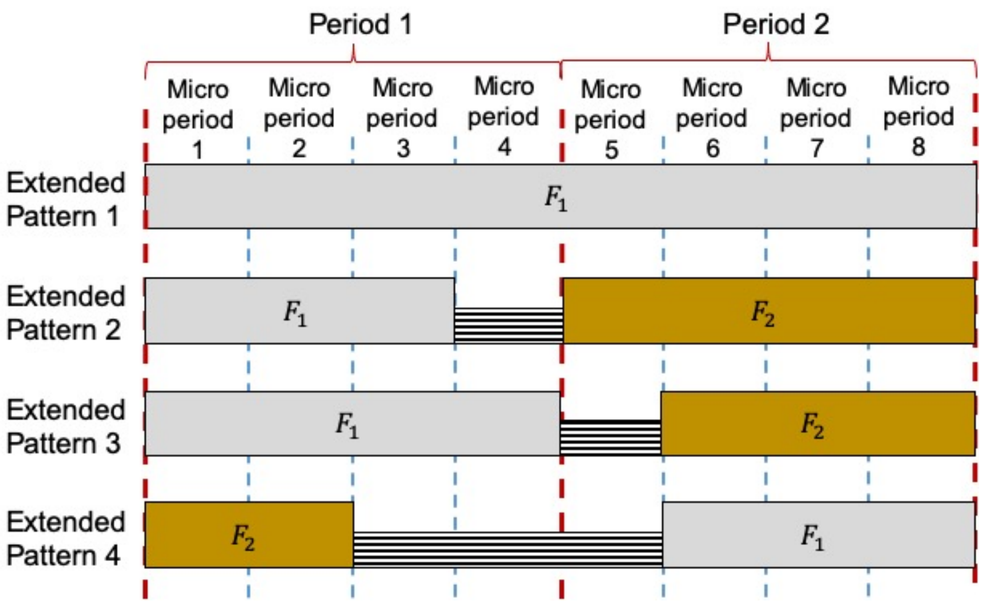

Figure 5.2. Valid extended patterns

5.4.2. Reformulated Mathematical Model

An extended pattern covers the entire planning horizon by definition, and assuming such patterns will be constructed ensuring the minimum production duration of families and setup transition feasibility, it enables us to simplify FTBMV-PM. We can associate each production line with an *extended* pattern for the entire planning horizon, hence reformulate the campaign planning problem. We name the reformulation as Extended Pattern Based Campaign Planning Model (CPM-EP). Table 5.4 lists the symbols used in CPM-EP along with their brief descriptions.

Table 5.4. Symbols used in CPM-EP.

Set	Description
J	Set of products
R	Set of production lines
Q	Set of quality groups
S	Set of size groups
T	Set of time periods
P	Set of extended campaign patterns
F	Set of product families
$P(f)$	Set of patterns containing family f at least once
$F(p)$	Set of families belonging to pattern p
$J(f)$	Set of products belonging to family f
Parameter	**Description**
D_{jt}	Demand of product j in period t
$I_{j(-1)}$	Beginning inventory of product j
v_{jr}	Production speed of product j on production line r
$S(j)$	Index of the size group of product j
$Q(j)$	Index of the quality group of product j
R_{fqsr}	Maximum production ratio for quality group q and size group s for family f on production line r
d_{fpt}	Number of days family f appears in extended pattern p in period t

Table 5.4. Symbols used in CPM-EP. (cont.)

Parameter	**Description**
h_j	Inventory holding cost for product j
b_j	Cost of backlogging a demand of product j for a single period
u_{jr}	Unit production cost for producing product j on line r
c_{pr}	Total setup cost for family order within extended pattern p on production line r
Variable	**Description**
I_{jt}	Inventory of product j at the end of period t
S_{jtk}	Satisfied quantity of demand from period t of product j in period k
U_{jt}	Unsatisfied quantity of demand from period t of product j
X_{jrt}	Production quantity of product j on production line r in period t
δ_{pr}	Binary indicator variable for selection of extended pattern p on production line r

Model 5.3. *Extended Pattern Based Campaign Planning Model (CPM-EP)*

$$\textit{Minimize} \quad \sum_{j \in J} \sum_{t \in T} \left[h_j \; I_{jt} + b_j \; (|T| - t + 1) \; U_{jt} + \sum_{\substack{k \in T \\ k \leq t}} (b_j \; (t-k) \; S_{jkt}) \right] + \sum_{r \in R} \sum_{j \in J} \sum_{t \in T} u_{jr} X_{jrt} + \sum_{r \in R} \sum_{p \in P} c_{pr} \delta_{pr}$$

subject to

$$\sum_{\substack{k\in T\\k\geq t}} S_{jtk} + U_{jt} = D_{jt} \qquad \forall\, j \in J,\ t \in T \tag{5.28}$$

$$I_{j(t-1)} + \sum_{r\in R} X_{jrt} - \sum_{\substack{k\in T\\k\leq t}} S_{jkt} = I_{jt} \qquad \forall\, j \in J,\ t \in T \tag{5.29}$$

$$\sum_{\substack{j\in J(f)\\Q(j)\leq q\\S(j)\leq s}} X_{jrt} \leq \sum_{j\in J(f)} X_{jrt}\ R_{fqrs} \qquad \forall\, r \in R,\ f \in F,\ q \in Q,\ s \in S,\ t \in T \tag{5.30}$$

$$\sum_{p\in P} \delta_{pr} = 1 \qquad \forall\, r \in R \tag{5.31}$$

$$\sum_{j\in J(f)} v_j X_{jrt} - \sum_{p\in P(f)} d_{fpt}\delta_{pr} = 0 \qquad \forall\, f \in F,\ r \in R,\ t \in T \tag{5.32}$$

$$I_{jt}, U_{jt} \geq 0 \quad \forall (j,t) \tag{5.33}$$

$$S_{jtk} \geq 0 \quad \forall (j,t,k \geq t) \tag{5.34}$$

$$X_{jrt} \geq 0 \quad \forall (j,r,t) \tag{5.35}$$

$$\delta_{pr} \in \{0,1\} \quad \forall (p,r) \tag{5.36}$$

Model 5.3 represents the complete formulation of CPM-EP. The objective is the same cost minimization with the exception that it has fewer terms since there is no need to represent the setup costs incurred separately for within period and over period boundaries as in Model 5.2. Equations (5.28)–(5.30) are requirement balance, inventory balance and size group quality constraints exactly the same as in Model 5.2. Eq. (5.31) ensures only a single extended pattern is assigned to a production line. Eq. (5.32) couples production quantity variables (X) with designated duration of corresponding families in the selected extended pattern, ensuring the plan respects the capacity of each production line. Finally, Equations (5.33)–(5.36) define variable domains.

Note that Model 5.3 is a reformulation of Model 5.2 with only five sets of constraints and variable bounds. Moreover, Model 5.3 has an exponential number of variables making it a candidate for column generation (CG) approach.

5.4.3. Column Generation

The main decision in CPM-EP is the assignment of an extended pattern to machines, and the number of extended patterns for each machine varies according to the number of families that can be produced on the machine, their respective minimum production durations and feasibility of setups between families. Hence, there can be an exponential number of extended patterns, which means in an optimal solution to CPM-EP, most of the corresponding δ variables will be equal to zero. As a first step of our column generation strategy, we relax the binary variables, δ, in CPM-EP and obtain the linear programming relaxation of restricted master problem as follows:

Model 5.4. *Restricted Extended Pattern Based Campaign Planning Master Model (RCPM-EP)*

$$
\begin{aligned}
min \quad & \sum_{j \in J} \sum_{t \in T} \left[h_j \; I_{jt} + b_j \; (|T| - t + 1) \; U_{jt} + \sum_{\substack{k \in T \\ k \leq t}} (b_j \; (t-k) \; S_{jkt}) \right] \\
& + \sum_{r \in R} \sum_{j \in J} \sum_{t \in T} u_{jr} X_{jrt} + \sum_{r \in R} \sum_{p \in P'} c_{pr} \delta_{pr}
\end{aligned}
$$

subject to

$$(5.28)\text{–}(5.32) \tag{5.37}$$

$$I_{jt}, U_{jt} \geq 0 \quad \forall (j,t) \tag{5.38}$$

$$S_{jtk} \geq 0 \quad \forall (j,t,k \geq t) \tag{5.39}$$

$$X_{jrt} \geq 0 \quad \forall (j,r,t) \tag{5.40}$$

$$0 \leq \delta_{pr} \leq 1 \quad \forall (p,r), \quad p \in P' \tag{5.41}$$

RCPM-EP considers a subset of extended patterns denoted with P'. To generate columns which are not already in P', we are interested in the reduced cost value associated with each potential extended pattern to be added to P'.

By definition, reduced cost is the amount of necessary improvement in the objective coefficient of the corresponding variable so that the variable becomes a basic variable. Moreover, reduced cost can be calculated by using optimal dual multipliers of the master problem.

We denote the dual variables associated with Equations (5.31) and (5.32) by π_r and μ_{frt} respectively. We can find a new column, namely an extended pattern, by checking its reduced cost such that reduced cost with respect to Eq. (5.42) is minimized,

$$\bar{\pi}_r - \sum_{f \in P(f)} \sum_{t \in T} d_{fpt}\bar{\mu}_{frt} \leq c_{pr} \quad \forall\, r \in R \tag{5.42}$$

where $\bar{\pi}_r$ and $\bar{\mu}_{frt}$ are optimal dual multipliers from RCPM-EP. Given an optimal solution of RCPM-EP, pricing subproblem (SP($\bar{\pi}_r$,$\bar{\mu}_{frt}$)) can be formulated as follows:

Model 5.5. *Pricing Subproblem (SP($\bar{\pi}_r$,$\bar{\mu}_{frt}$))*

$$\begin{aligned} &\textit{Minimize } c_{pr} - \bar{\pi}_r + \sum_{f \in P(f)} \sum_{t \in T} d_{fpt}\bar{\mu}_{frt} \\ &\textit{subject to } p \in P \\ &\qquad\qquad\quad d_{fpt} \in \mathbb{Z} \end{aligned}$$

5.4.4. Pricing Subproblem as Shortest Path Problem

We define the pricing problem with Model 5.5 for each production line r, with $\bar{\pi}_r$ and $\bar{\mu}_{frt}$ being parameters to (SP($\bar{\pi}_r$,$\bar{\mu}_{frt}$)) as optimal dual multipliers from RCPM-EP. The aim of pricing problem is to construct an extended pattern such that the minimum production duration of each family f is respected, the transitions, namely setups, between families are feasible and its associated reduced cost is as small as possible.

We can represent an extended pattern as a path on a special network of the corresponding machine. For each micro period in the planning horizon we create a node for each family f. Hence, when a node is in a path, it means that the production line r is dedicated to producing products of the corresponding family f in that micro period. In addition, we create a source and a sink node so that a path maps to a directed flow between them. Arcs of this network is constructed in a way that:

- the minimum production duration of each family is respected
- there exist arcs between family pairs such that a setup is feasible
- when an arc corresponds to a setup, it respects both the setup duration in between families and the minimum duration of the successor family.

Note that, for arcs corresponding to a setup and minimum duration of the successor family, nodes of the family in related *micro periods* are not in the path but they are in the production plan.

In order to illustrate the idea, let F_1 and F_2 be two families to be produced on a machine in a planning horizon of 8 micro periods, with 2 micro periods of minimum production duration each. Sequence-dependent setup times from F_1 to F_2 and from F_2 to F_1 are 1 and 3 micro periods respectively. We further assume that, each period consists of four micro periods. Figure 5.3 shows the corresponding network.

All the paths in this network are valid extended patterns. To further clarify the illustration, we provide in Figure 5.4 the gantt representation of the path consisting of arcs in dotted lines. Arc covering micro periods 3 to 7 corresponds to a setup from F_1 to F_2 on micro periods 3 to 5, and minimum duration of F_2 on micro periods 6 and 7.

Considering our motivation to generate a new column for RCPM-EP with a promising reduced cost defined as the objective function of Model 5.5, it is sufficient for us to calculate proper costs on arcs. For outbound arcs of source and inbound arcs of sink node, the associated cost is zero. For all other arcs, the cost is calculated as follows:

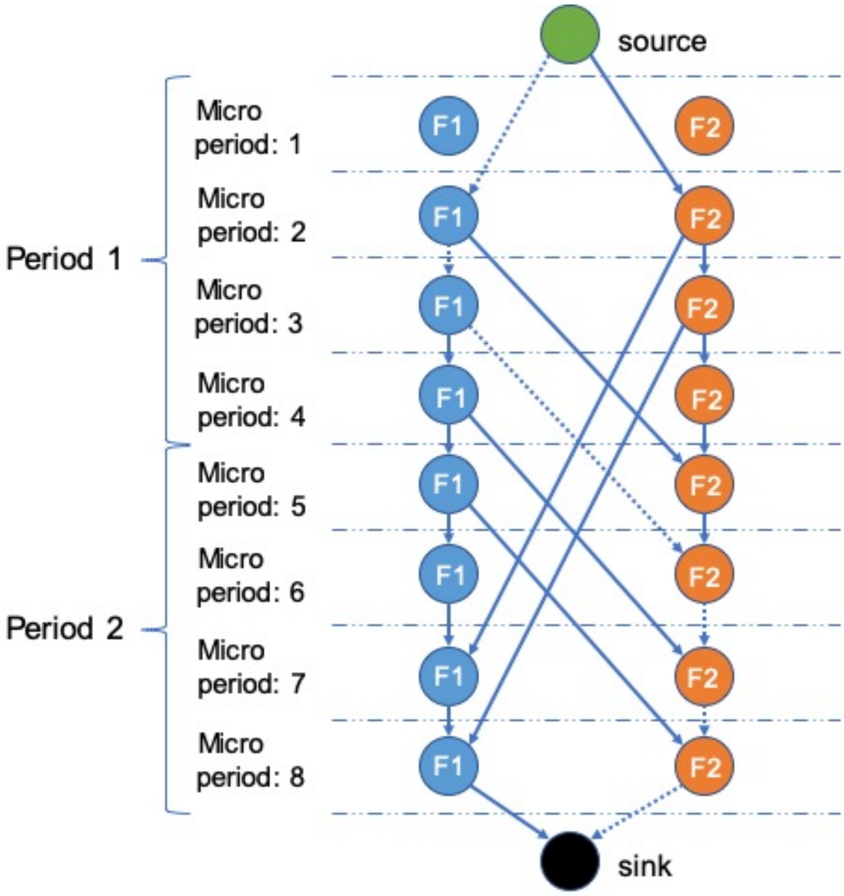

Figure 5.3. An illustrative s-t network

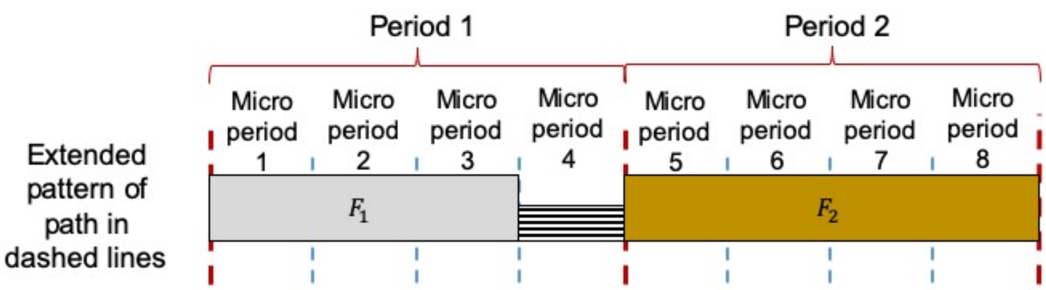

Figure 5.4. Illustration of an s-t network path

we determine the duration of production in each period and multiply with the proper dual multiplier of the family associated with *to node*. Moreover, if family associated with *from node* is different than the one associated with *to node*, we account for a setup cost and the minimum duration of this *destination* family. Once we calculate the arc costs, the most promising candidate extended pattern is given by the s-t shortest path in this network.

We note that the network is a directed acyclic graph (DAG). Since there is no cycle in a DAG by definition, no negative cost cycle can exist. Hence, shortest paths are well defined, and we can solve it efficiently with topological ordering, in $O(|A|)$ time complexity following from [54], where $|A|$ is the number of arcs in the network. Pricing subproblem (SP($\bar{\pi}_r$,$\bar{\mu}_{frt}$)) can be represented as a DAG and at each iteration we can generate a new column, namely an extended pattern, in polynomial time by calculating proper arc costs.

5.4.5. Algorithm Details

We will apply a B&P algorithm, which focuses on generating columns for tightening an LP relaxation, for the solution of CPM-EP. We defined column generation and pricing problem as shortest path problem in Section 5.4.3 and Section 5.4.4 respectively. In this section, we will focus on the details of the algorithm.

5.4.5.1. Generating Initial Set of Columns. Starting the algorithm requires an initial set of columns to be assumed as P' that RCPM-EP will run with. In principal, such a subset can be determined by running constructive heuristics. We generate "unit" extended patterns, which are patterns such that only a single family f is produced on a production line during the entire planning horizon. We generate all possible unit patterns P_u for all resources and start the B&P algorithm using them as initial set of columns.

5.4.5.2. Branching and Node Selection Strategy. At each iteration, we solve RCPM-EP with P', followed by solving the pricing subproblem to identify columns to enter the basis for improved objective. Note that we solve the pricing s-t shortest path problem for each production line and add all new columns to P'. When the pricing problem is unable to generate a column that will price out, the solution to RCPM-EP is optimal if it is integer feasible. Otherwise, this means some δ variables are fractional and we need to do branching.

There are fundamental difficulties in applying column generation techniques for linear programming in integer programming solution methods [50]. It is essential to choose a branching rule which does not increase the complexity of pricing problem solution. Considering conventional branching on variables, it has the potential to destruct the structure of the pricing problem, which is the case for our network representation. Suppose that we obtain a fractional value in the optimal solution to RCPM-EP on the illustrative network in Figure 5.3 for the extended pattern p'. This p' is composed as $F_1 - F_1 - F_1 - SETUP - F_2 - F_2 - F_2 - F_2$. In order to branch on corresponding δ variable, for the upper branch we can remove all the arcs other than representing p' from the network of the production line. However, for the lower branch, to have $\delta_{p'} = 0$, if we remove the arcs not included in p', then we also cut off some other feasible extended patterns as well, of which $F_1 - F_1 - F_1 - F_1 - F_1 - F_1 - F_1 - F_1$ is an example in the example on Figure 5.3. Such branching destructs the structure of the pricing problem since we will need a special algorithm, which in the worst case requires enumerating all the paths leading to exponential complexity in $|N|$. Hence, we need to adopt another branching rule. An intuitive approach is to define a branching strategy that will correspond to removal of some arcs and/or nodes from the network.

We define a branching strategy based on family nodes in the network of each production line r. When the CG is unable to generate new columns, we calculate, for each arc on the network, a weight index such that for each arc we calculate the sum of all the extended pattern variables with positive value. This provides us a "superposition" of these patterns. Then starting from the source node, we start tracking paths.

At some node, the paths will need to be separated which provides reasonable branching point on the network.

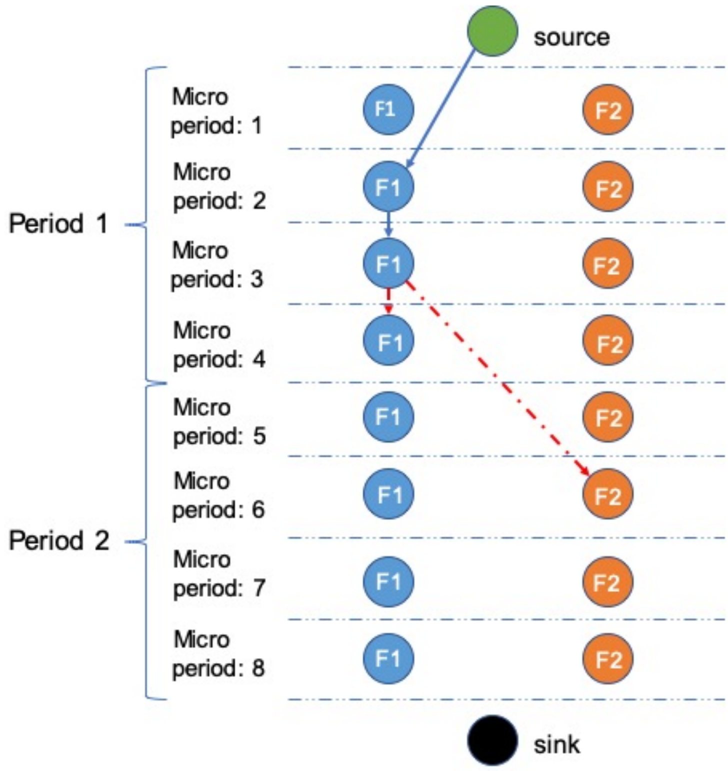

Figure 5.5. An illustrative branching instance

Figure 5.5 illustrates such an instance. There are two different paths and each one reaches F_1 in micro period 2 from source, continues on F_1 onto micro period 3. At the end of micro period 3, there are two possible next nodes to reach: F_1 in micro period 4 and F_2 in micro period. Suppose that weights on corresponding arcs are 0.2 and 0.8. We select the arc with highest value, which in this case is the arc from F_1 in micro period 3 to F_2 in micro period 6. Our branching decision is then on upper branch, the network should have to provide a path being in F_2 producing state in micro period 4 from now on throughout the algorithm. On the lower branch, the network should not provide any path being in F_2 producing state in micro period 4.

Note that we do the branching in the adjacent micro period of the last common node and that being in the setup for destination family f is admitted as producing f, in this case F_2. Moreover, we add B&P node of single production line at each iteration.

This is to avoid redundant search in the tree. We select the resource to create the branches such that the resource has its separation at the earliest micro period. If there are multiple resource separation at same micro period, then we select the resource with highest number of families that can be produced on.

At each node, we simply activate or deactivate a node from the network corresponding to a family in a micro period, which we call the branching rule. In practical terms, this corresponds to deciding whether to produce a family in a specific micro period on a production line or not. Considering the branching instance described above, we can intuitively remove the node F_2 in micro period 4 for the lower branch so that the machine produces F_1. However, this approach is incomplete since there exist arcs which implicitly adds up to the undesired state. In Figure 5.3 which is the complete network, arc from F_1 in micro period 2 to node F_2 in micro period 5 means that the machine is producing F_2 in micro period 4. Hence, it requires to carefully account for all nodes and arcs and remove as necessary to ensure the desired state of the machine on the lower and upper branches.

Dashed arcs (in green) in Figure 5.6 means the machine will produce F_2 in micro period 4 and dotted arcs (in red) impose the opposite. Solid arcs are unrelated to the machine state in micro period 4. Figure 5.7 shows the residual network for upper branch in (a) and for lower branch in (b). Note that, we also removed F_1 node in the upper and F_2 node in the lower branch along with their inbound and outbound arcs.

Since we employ all the related branching rules to the network at each B&P node to be processed , any generated column throughout the algorithm respects all of these rules. Hence, going down the B&P search tree, we will not get any infeasibility and will work on more and more sparse s-t networks. We apply three most common strategies for the node selection from the search tree: depth-first-search (DFS), breadth-first-search (BFS) and best-bound-selection (BBS).

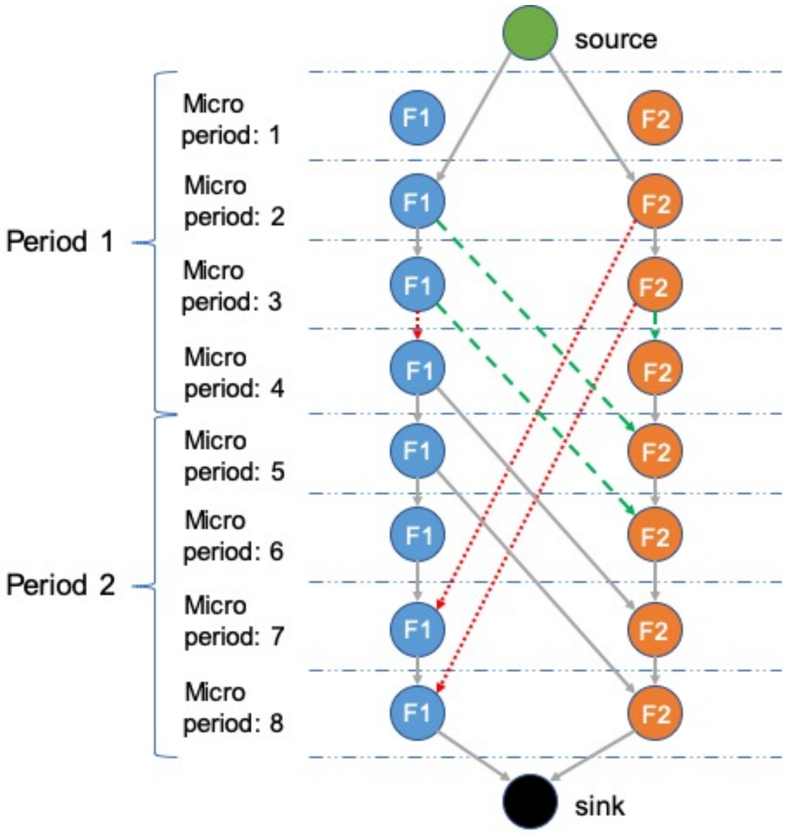

Figure 5.6. Color coded arcs related to producing F_2 in micro period 4

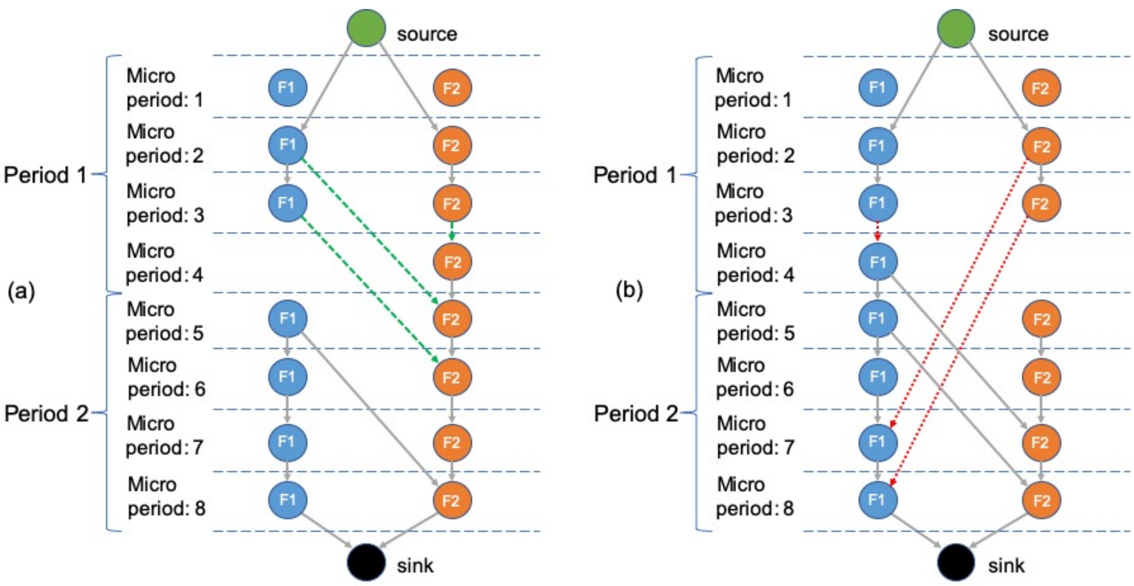

Figure 5.7. Upper (a) and Lower (b) branches related to producing F_2 in micro period 4

5.4.5.3. Upper Bound Generation. Upper bound generation is an essential part of B&P algorithms in the sense that it potentially provides an incumbent solution leads to pruning of branches yielding lower bounds higher than the upper bound. Hence, it is necessary to come up with an improved approach such that it is capable of balancing running time efficiency with quality of the solutions found.

As a heuristic approach we propose to run the CPM-EP with P' such that P' contains all generated columns if we are processing the root node. Otherwise, P' will consist of extended patterns with positive δ variable value in the latest solution of RCPM-EP. In this way, the heuristic can search within a larger set of columns while processing the root node and a limited subset while processing any other node.

We limit the execution time of CPM-EP models by 300 seconds for each execution of our heuristic. Since the model in the root node contains, in general, a much larger set of columns compared to other nodes, it might terminate without proven optimality. The CPM-EP on other nodes in the B&P tree reaches to an optimal solution within 30 seconds according our preliminary observations.

5.4.5.4. Adaptive Root Node Processing. Root node is by definition the first node in the search tree. Intuitively, we start with P' generated as described in Section 5.4.5.1. However, using *unit* extended patterns as P' is questionable in the sense that the CG algorithm might converge slowly to the state where it cannot generate any column. It can be possible for CG to converge faster with a different initial set of P'. Making use of micro-period concept can provide another means of generating initial set of patterns such that the root node processing performance is better with respect to processing time.

We defined micro-periods as a unit amount of time multiplies of which can represent problem data, especially the input data in continuous resolution. The longer the micro-period length is, the more lightweight will be the shortest path network structure. Consequently, the CG algorithm may converge faster.

On the other hand, having longer micro-periods will result in discrepancy in the sense that the input data is not incorporated into the model fully. We suggest an approach to dynamically change the micro-period length throughout the processing of root node, which we call *adaptive micro-period length*.

We set an initial positive integer k value, and for each k, we process the root node setting the micro-period length to 2^k unit amount of time. Note that, we use unit extended patterns as P' for the initial k value. Once CG converges, we keep the incumbent solution in P', remove all other patterns, reduce k by 1 and iterate on until $k = 0$. For each value of k, the network is a *'shrunk'* version of the one with $k + 1$. Hence, an extended pattern, a path in the network context, can be represented in either of the networks. Note also that, for $k = 0$, we have the original problem but this time with a different initial column set than unit extended patterns. This structure allows us to adaptively change the sensitivity of the model to input data and obtain initial solutions that will help CG convergence faster.

6. MULTIPLE LEVEL NETWORK PROBLEM

In this chapter, we extend the parallel machine problem defined in Chapter 5 to multiple levels and to network structure. As defined in Chapter 3, float glass is used to produce mirrored, coated, and laminated glass, which all have discrete production lines as opposed to float glass. Hence, there exists a bill of material (BOM) for such products. Moreover, some coated glass are produced from laminated glass, which is produced from float. A product is consumed to satisfy its own customer demand as well as a component for downstream products.

For the network consideration, we introduce facilities that contain the production lines. Moreover, we also allow customers to exist within the network as locations. To satisfy a demand at a customer location, we introduce transportation between locations. Figure 6.1 illustrates a network with three facilities denoted as $L1, L2$ and $L3$, and two customer locations denoted as $C1$ and $C2$. Blue circles correspond to float glass production lines whereas orange pentagons correspond to secondary discrete production lines. Moreover, arrows stand for transportation options. Note that, a product can be shipped from different facilities to the same customer as a result of the campaign plan. The decision depends on the production costs on each production line in locations as well as the corresponding transportation cost.

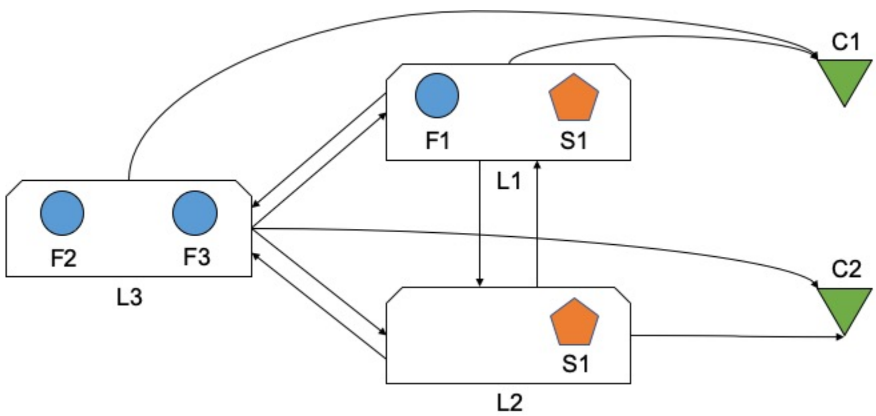

Figure 6.1. Illustration of multiple level network

We extend FTBMV-PM and B&P methods formulated in Chapter 5. Note that we do not further extend the PTBMV-PM since it is outperformed by other methods in both single and parallel machine instances. Following some preliminary tests, we design mathematical programming based heuristic algorithms for this multiple level network variant of the campaign planning problem.

6.1. Family Transition Based Model Variant on Multiple Level Network

In this Section, we present the multiple facility multiple level extension of FTBMV, namely FTBMV-MLN. Table 6.1 lists the symbols used in FTBMV-MLN. We present constraints in Section 5.3.1 and complete model with objective in Section 5.3.2.

Table 6.1. Symbols used in FTBMV-MLN.

Set	**Description**
J	Set of products
$J'(j)$	Set of products that consume product j
L	Set of facilities
R	Set of production lines
$R(l)$	Set of production lines in facility l
R^C	Set of continuous production lines
R^D	Set of discrete production lines
Q	Set of quality groups
S	Set of size groups
T	Set of time periods
P	Set of campaign patterns
F	Set of product families
O	Set of orders for timing of production in a period
	(b: beginning, m: middle, e: end)
$P(f)$	Set of patterns containing family f at least once
$F(p)$	Set of families belonging to pattern p
$F^o(p)$	Set of families appearing in order o in pattern p
$J(f)$	Set of products belonging to family f

Table 6.1. Symbols used in FTBMV-MLN. (cont.)

$\Gamma^r(f,g)$	Set of product family couples that are infeasible on production line r, $f,g \in F$
$P^S(f)$	Patterns that family f is the first family
$P^E(f)$	Patterns that family f is the last family
Parameter	**Description**
D_{jlt}	Demand of product j in facility l in period t
$I_{jl(-1)}$	Beginning inventory of product j in facility l
v_{jr}	Production speed of product j on production line r
$k_{jj'}$	Amount of product j consumed to produce one unit of product j', $j' \in J'(j)$
A_t	Available capacity of production lines in period t
$S(j)$	Index of the size group of product j
$Q(j)$	Index of the quality group of product j
R_{fqsr}	Maximum production ratio for quality group q and size group s for family f on production line r
MD_{fr}	Minimum production duration for family f on production line r
NT_{fp}	Number of times family f appears in the middle order of pattern p
ST_{pr}	Setup time needed for family order within pattern p on production line r
ST_{fgr}	Setup time needed for switching from product family f to family g on production line r
h_{jl}	Inventory holding cost for product j in facility l
b_{jl}	Cost of backlogging a demand of product j in facility l for a single period
u_{jr}	Unit production cost for producing product j on line r
n_{jlm}	Unit transportation cost of product j from facility l to facility m
c_{fgr}	Setup cost for switching from family f to family g on production line r
c_{pr}	Total setup cost of pattern p on production line r

Table 6.1. Symbols used in FTBMV-MLN. (cont.)

Variable	**Description**
I_{jlt}	Inventory of product j in facility l at the end of period t
S_{jltk}	Satisfied quantity of demand of product j in facility l from period t in period k
U_{jlt}	Unsatisfied quantity of demand of product j in facility l from period t
X_{jrt}	Production quantity of product j on production line r in period t
T_{jlmt}	Transported quantity of product j from facility l to facility m in period t
δ_{prt}	Binary indicator variable for selection of pattern p on production line r in period t
d^o_{frt}	Number of days spent for production of family f in order o on production line r in period t
γ^S_{frt}	Indicator for selection of family f as starting on production line r in period t
γ^E_{frt}	Indicator for selection of family f as ending on production line r in period t
θ_{fgrt}	Auxiliary variable indicating whether production line r switched from family f to family g at the beginning of period t
n^P_{fgrt}	Number of days spent for setup on production line r in predecessor period $t-1$ for switched from family f to family g at the beginning of period t
n^S_{fgrt}	Number of days spent for setup on production line r in successor period t for switched from family f to family g at the beginning of period t
F_{rt}	Setup time spent on production line r at the beginning of period t
B_{rt}	Setup time spent on production line r at the end of period t

6.1.1. Constraints

Eq. (6.1) ensures the consistency of demand satisfactions having demand backlog and unsatisfaction allowed. Note that, we introduced the index l to represent locations.

$$\sum_{\substack{k\in T\\ k\geq t}} S_{jltk} + U_{jlt} = D_{jlt} \qquad \forall\, j \in J,\ l \in L,\ t \in T \tag{6.1}$$

Eq. (6.2) is the inventory balance constraint. It links the production quantity X, transportation T inventory I and demand satisfaction S variables across time periods. Also note that, consumption of the inventory by other products through consumption rate k.

$$I_{jl(t-1)} + \sum_{r\in R(l)} X_{jrt} + \sum_{\substack{m\in L\\ m\neq l}} T_{jmlt} - \sum_{\substack{k\in T\\ k\leq t}} S_{jlkt} - \sum_{j'\in J'(j)} \sum_{r\in R(l)} k_{jj'} X_{j'rt} - \sum_{\substack{m\in L\\ m\neq l}} T_{jlmt} = I_{jlt} \qquad \forall\, j \in J,\ l \in L,\ t \in T \tag{6.2}$$

Eq. (6.3) couples variables representing number of days of production allocated in an order for a family to production quantity variables. Note that, we only have such constraints for continuous production lines.

$$\sum_{j\in J} v_{jr} X_{jrt} = \sum_{o\in O} d^{o}_{frt} \qquad \forall\, f \in F,\ r \in R^{C},\ t \in T \tag{6.3}$$

Eq. (6.4) formulates production line capacity for continuous, and Eq. (6.5) for discrete production lines.

$$\sum_{f\in F} \sum_{o\in O} d^{o}_{frt} + \sum_{p\in P} ST_{pr}\delta_{prt} + F_{rt} + B_{rt} = A_t \qquad \forall\, r \in R^{C},\ t \in T \tag{6.4}$$

$$\sum_{j\in J} v_{jr} X_{jrt} \leq A_t \qquad \forall\, r \in R^{D},\ t \in T \tag{6.5}$$

Eq. (6.6) ensures that production schedule consists of a feasible composition in terms of size group and quality for a continuous production line.

$$\sum_{\substack{j \in J(f) \\ Q(j) \leq q \\ S(j) \leq s}} X_{jrt} \leq \sum_{j \in J(f)} X_{jrt} \; R_{fqrs} \qquad \forall \; r \in R^C, \; f \in F, \; q \in Q, \; s \in S, \; t \in T \tag{6.6}$$

Eq. (6.7) allocates patterns to continuous production lines for each period.

$$\sum_{p \in P} \delta_{prt} = 1 \qquad \forall \; r \in R^C, \; t \in T \tag{6.7}$$

Equations (6.8)–(6.10) serve for modeling a lower bound for duration of families that are produced in the middle of a pattern, split into two adjacent periods and a proper upper bound respectively.

$$d^m_{frt} \geq MD_{fr} NT_{fp} \delta_{pt} \qquad \forall \; r \in R^C, \; p \in P, \; f \in F^m(p), \; t \in T \tag{6.8}$$

$$d^e_{fr(t)} + d^b_{fr(t+1)} \geq MD_{fr} \delta_{prt} \qquad \forall \; r \in R^C, \; p \in P, \; f \in F^b(p) \cup F^e(p), \; t \in T \tag{6.9}$$

$$d^o_{frt} \leq \sum_{p \in P^o(f)} A_t \delta_{prt} \qquad \forall \; r \in R^C, \; f \in F, \; o \in O, \; t \in T \tag{6.10}$$

Equations (6.11)–(6.12) determine starting and ending family within a period for continuous production lines. Similar to FTBMV-PM, we can relax (γ^S, γ^E) as $\gamma^S, \gamma^E \geq 0$.

$$\gamma^S_{frt} = \sum_{p \in P^S(f)} \delta_{prt} \qquad \forall \; r \in R^C, \; f \in F, \; t \in T \tag{6.11}$$

$$\gamma^E_{frt} = \sum_{p \in P^E(f)} \delta_{prt} \qquad \forall \; r \in R^C, \; f \in F, \; t \in T \tag{6.12}$$

Equations (6.13)–(6.15) are the counterpart of Equations (5.21)–(5.23), formulating θ variables for correct color transition management.

$$\theta_{fgrt} \leq \gamma^E_{fr(t-1)} \qquad \forall\, r \in R^C,\ f,g \in F,\ t \in T,\ t \geq 1 \tag{6.13}$$

$$\theta_{fgrt} \leq \gamma^S_{grt} \qquad \forall\, r \in R^C,\ f,g \in F,\ t \in T \tag{6.14}$$

$$\theta_{fgrt} \geq \gamma^E_{fr(t-1)} + \gamma^S_{grt} - 1 \qquad \forall\, r \in R,\ f,g \in F,\ t \in T,\ t \geq 1 \tag{6.15}$$

Eq. (6.16) ensures that each continuous production line allocates necessary setup time for color transition.

$$n^P_{fgrt} + n^S_{fgrt} = ST_{fgr}\theta_{fgrt} \qquad \forall\, r \in R^C,\ f,g \in F,\ (f,g) \notin \Gamma^r(f,g),\ t \in T \tag{6.16}$$

Equations (6.17)–(6.18) relate setup time variables for families (n^S, n^E) to period based variables (F, B).

$$F_{rt} = \sum_{(f,g) \notin \Gamma^r(f,g)} n^S_{fgrt} \qquad \forall\, r \in R^C,\ f,g \in F,\ t \in T \tag{6.17}$$

$$B_{rt} = \sum_{(f,g) \notin \Gamma^r(f,g)} n^P_{fgr(t+1)} \qquad \forall\, r \in R^C,\ f,g \in F,\ t \in T \tag{6.18}$$

Eq. (6.19) is the counterpart of Eq. (5.27). to avoid infeasible family transitions.

$$\sum_{\substack{f,g \in F \\ (f,g) \notin \Gamma^r(f,g)}} \theta_{fgrt} = 1 \qquad \forall\, r \in R^C,\ t \in T,\ t \geq 1 \tag{6.19}$$

6.1.2. Objective and Complete Model

The objective function, similar to all previous instances, is minimizing demand satisfaction, inventory holding, production and setup costs. Additionally FTBMV-MLN includes transportation costs. Model 6.1 represents the complete formulation.

Model 6.1. *Family Transition Based Model Variant on Multiple Level Network (FTBMV-MLN)*

$$\begin{aligned}
\textit{Minimize } & \sum_{j\in J}\sum_{l\in L}\sum_{t\in T}\left[h_{jl}\ I_{jlt}+b_{jl}\ (|T|-t+1)\ U_{jlt}+\sum_{\substack{k\in T\\ k\le t}}(b_j\ (t-k)\ S_{jlkt})\right]\\
&+\sum_{r\in R}\sum_{j\in J}\sum_{t\in T}u_{jr}X_{jrt}+\sum_{j\in J}\sum_{l\in L}\sum_{\substack{m\in L\\ m\neq l}}\sum_{t\in T}n_{jlm}T_{jlmt}\\
&+\sum_{r\in R^C}\sum_{p\in P}c_{pr}\delta_{pr}+\sum_{r\in R^C}\sum_{t\in T}\sum_{(f,g)\notin\Gamma^r(f,g)}c_{fgr}\theta_{fgrt}
\end{aligned}$$

subject to (6.1)–(6.19)

$$\begin{aligned}
& I_{jlt}, U_{jlt}\geq 0 \qquad \forall(j,l,t)\\
& S_{jltk}\geq 0 \qquad \forall(j,l,t,k\geq t)\\
& d^o_{frt}\geq 0 \qquad \forall(f,r\in R^C,t)\\
& X_{jrt}\geq 0 \qquad \forall(j,r,t)\\
& T_{jlmt}\geq 0 \qquad \forall(j,l,m\neq l,t)\\
& \delta_{prt}\in\{0,1\} \qquad \forall(p,r\in R^C,t)\\
& 0\leq \theta_{fgrt} \qquad \forall(f,g,r\in R^C,t)\\
& \gamma^S_{frt},\gamma^E_{frt}\geq 0 \qquad \forall(f,r\in R^C,t)\\
& F_{rt}, B_{rt}\geq 0 \qquad \forall(r\in R^C,t)\\
& n^P_{fgrt}, n^S_{fgrt}\geq 0 \qquad \forall(f,g,r\in R^C,t)
\end{aligned}$$

6.2. Branch-and-Price Algorithm on Multiple Level Network

In this Section, we present the multiple facility multiple level extension of the B&P algorithm presented in Section 5.4. Recall that the main driver of the CG and B&P algorithm is the concept of *extended* pattern def in Section 5.4.1. We observe that it is still applicable for this extension of the problem, which lets us to adapt also the reformulated mathematical model CPM-EP for multiple level network. We name this new model CPM-EPMLN. Table 6.2 lists the symbols used in CPM-EPMLN along with their brief descriptions.

Table 6.2. Symbols used in CPM-EPMLN.

Set	**Description**
J	Set of products
$J'(j)$	Set of products that consume product j
L	Set of facilities
R	Set of production lines
$R(l)$	Set of production lines in facility l
R^C	Set of continuous production lines in facility l
R^D	Set of discrete production lines in facility l
Q	Set of quality groups
S	Set of size groups
T	Set of time periods
P	Set of extended campaign patterns
F	Set of product families
$P(f)$	Set of patterns containing family f at least once
$F(p)$	Set of families belonging to pattern p
$J(f)$	Set of products belonging to family f

Table 6.2. Symbols used in CPM-EPMLN. (cont.)

Parameter	Description
D_{jt}	Demand of product j in period t
$I_{j(-1)}$	Beginning inventory of product j
v_{jr}	Production speed of product j on production line r
$k_{jj'}$	Amount of product j consumed to produce one unit of product j', $j' \in J'(j)$
$S(j)$	Index of the size group of product j
$Q(j)$	Index of the quality group of product j
R_{fqsr}	Maximum production ratio for quality group q and size group s for family f on production line r
d_{fpt}	Number of days family f appears in extended pattern p in period t
h_j	Inventory holding cost for product j
b_j	Cost of backlogging a demand of product j for a single period
u_{jr}	Unit production cost for producing product j on line r
n_{jlm}	Unit transportation cost of product j from facility l to facility m
c_{pr}	Total setup cost for family order within extended pattern p on production line r
Variable	**Description**
I_{jt}	Inventory of product j at the end of period t
S_{jtk}	Satisfied quantity of demand from period t of product j in period k
U_{jt}	Unsatisfied quantity of demand from period t of product j
X_{jrt}	Production quantity of product j on production line r in period t
T_{jlmt}	Transported quantity of product j from facility l to facility m in period t
δ_{pr}	Binary indicator variable for selection of extended pattern p on production line r

Model 6.2. *Extended Pattern Based Campaign Planning Model on Multiple Level Network (CPM-EPMLN)*

$$\begin{aligned}
\textit{Minimize} \quad & \sum_{j\in J}\sum_{l\in L}\sum_{t\in T}\left[h_{jl}\ I_{jlt} + b_{jl}\ (|T|-t+1)\ U_{jlt} + \sum_{\substack{k\in T\\ k\leq t}}(b_j\ (t-k)\ S_{jlkt})\right] \\
& + \sum_{r\in R}\sum_{j\in J}\sum_{t\in T} u_{jr}X_{jrt} + \sum_{j\in J}\sum_{l\in L}\sum_{\substack{m\in L\\ m\neq l}}\sum_{t\in T} n_{jlm}T_{jlmt} \\
& + \sum_{r\in R^C}\sum_{p\in P} c_{pr}\delta_{pr}
\end{aligned}$$

subject to

$$\sum_{\substack{k\in T\\ k\geq t}} S_{jltk} + U_{jlt} = D_{jlt} \qquad \forall\ j\in J,\ l\in L,\ t\in T \tag{6.20}$$

$$\begin{aligned}
I_{jl(t-1)} + \sum_{r\in R(l)} X_{jrt} + \sum_{\substack{m\in L\\ m\neq l}} T_{jmlt} - \sum_{\substack{k\in T\\ k\leq t}} S_{jlkt} - \sum_{j'\in J'(j)}\sum_{r\in R(l)} k_{jj'}X_{j'rt} - & \\
\sum_{\substack{m\in L\\ m\neq l}} T_{jlmt} = I_{jlt} \qquad \forall\ j\in J,\ l\in L,\ t\in T &
\end{aligned} \tag{6.21}$$

$$\sum_{\substack{j\in J(f)\\ Q(j)\leq q\\ S(j)\leq s}} X_{jrt} \leq \sum_{j\in J(f)} X_{jrt}\ R_{fqrs} \qquad \forall\ r\in R^C,\ f\in F,\ q\in Q,\ s\in S,\ t\in T \tag{6.22}$$

$$\sum_{p\in P} \delta_{pr} = 1 \qquad \forall\ r\in R^C \tag{6.23}$$

$$\sum_{j\in J(f)} v_j X_{jrt} - \sum_{p\in P(f)} d_{fpt}\delta_{pr} = 0 \qquad \forall\ f\in F,\ r\in R^C,\ t\in T \tag{6.24}$$

$$\sum_{j\in J} v_{jr}X_{jrt} \leq A_t \qquad \forall\ r\in R^D,\ t\in T \tag{6.25}$$

$$I_{jlt}, U_{jlt} \geq 0 \qquad \forall(j,l,t) \tag{6.26}$$

$$S_{jltk} \geq 0 \qquad \forall(j,l,t,k \geq t) \tag{6.27}$$

$$X_{jrt} \geq 0 \quad \forall(j,r,t) \tag{6.28}$$

$$T_{jlmt} \geq 0 \qquad \forall(j,l,m \neq l,t) \tag{6.29}$$

$$\delta_{pr} \in \{0,1\} \quad \forall(p, r \in R^C) \tag{6.30}$$

Model 6.2 represents the complete formulation of CPM-EPMLN. The objective is the same as the objective of 5.3 with the addition of transportation cost. Equations (6.20)–(6.22) are requirement balance, inventory balance and size group quality constraints. Eq. (6.23) ensures only a single extended pattern is assigned to a production line. Eq. (6.24) couples production quantity variables (X) with designated duration of corresponding families in the selected extended pattern. Eq. (6.25) is the resource capacity constraint for discrete production lines. Finally, Equations (6.26)–(6.30) define variable domains.

Similar to building the CG as explained in Section 5.4.3, we relax the binary variables, δ, in CPM-EPMLN and obtain the linear programming relaxation of restricted master problem as follows:

Model 6.3. *Restricted Extended Pattern Based Campaign Planning Master Model on Multple Level Network (RCPM-EPMLN)*

$$\begin{aligned}
min \quad & \sum_{j \in J} \sum_{l \in L} \sum_{t \in T} \left[h_{jl}\ I_{jlt} + b_{jl}\ (|T| - t + 1)\ U_{jlt} + \sum_{\substack{k \in T \\ k \leq t}} (b_j\ (t-k)\ S_{jlkt}) \right] \\
& + \sum_{r \in R} \sum_{j \in J} \sum_{t \in T} u_{jr} X_{jrt} + \sum_{j \in J} \sum_{l \in L} \sum_{\substack{m \in L \\ m \neq l}} \sum_{t \in T} n_{jlm} T_{jlmt} \\
& + \sum_{r \in R^C} \sum_{p \in P} c_{pr} \delta_{pr}
\end{aligned}$$

subject to

$$(6.20)\text{–}(6.24) \tag{6.31}$$

$$I_{jlt}, U_{jlt} \geq 0 \qquad \forall(j,l,t) \tag{6.32}$$

$$S_{jltk} \geq 0 \qquad \forall(j,l,t,k \geq t) \tag{6.33}$$

$$X_{jrt} \geq 0 \quad \forall(j,r,t) \tag{6.34}$$

$$T_{jlmt} \geq 0 \qquad \forall(j,l,m \neq l,t) \tag{6.35}$$

$$0 \leq \delta_{pr} \leq 1 \quad \forall(p, r \in R^C), \quad p \in P' \tag{6.36}$$

RCPM-EPMLN considers a subset of extended patterns, P'. we need the reduced cost associated with extended patterns, which are not already in P'. We denote the dual variables associated with Equations (6.23) and (6.24) by π_r and μ_{frt} respectively.

We can generate an extended pattern minimizing the reduced cost expressed with Eq. (6.37)

$$\bar{\pi}_r - \sum_{f \in P(f)} \sum_{t \in T} d_{fpt} \bar{\mu}_{frt} \leq c_{pr} \quad \forall\, r \in R \tag{6.37}$$

where $\bar{\pi}_r$ and $\bar{\mu}_{frt}$ are optimal dual multipliers from RCPM-EPMLN. Given an optimal solution of RCPM-EPMLN, pricing subproblem (SP($\bar{\pi}_r$,$\bar{\mu}_{frt}$)) can be formulated with Model 6.4.

Model 6.4. *Pricing Subproblem (SP($\bar{\pi}_r$,$\bar{\mu}_{frt}$))*

$$\textit{Minimize } c_{pr} - \bar{\pi}_r + \sum_{f \in P(f)} \sum_{t \in T} d_{fpt} \bar{\mu}_{frt}$$

$$\textit{subject to } p \in P$$

$$d_{fpt} \in \mathbb{Z}$$

To solve the pricing subproblem, we formulated it as a shortest path problem in parallel machine instance, as explained in Section 5.4.4. We observe that the pricing problem is no different for multiple machine network extension. Hence, we adopt the same pricing problem approach. Moreover, regarding the details of the B&P algorithm, everything described in Section 5.4.5 is still applicable.

6.3. Demand Projection Heuristics

In this Section we develop a new matheuristic exploiting the definition of the multiple level network structure of the campaign planning problem. We observe that not only demands for float glass products from various customer locations need to be addressed to production lines in different facilities, but also dependent demands from secondary products needs to be addressed. Hence, if we can come up with an efficient allocation of demands to float production locations, then we reduce the problem to the parallel machine instance. We formulate the matheuristic by making use of this observation. We give details of the demand projection approach in Section 6.3.1. We explain the algorithm as well as some variations of the algorithm in Section 6.3.2.

6.3.1. Reducing Network with Demand Projection

Multiple level network extension of the campaign planning problem considers secondary products, which consume float glass as a semi-product and multiple locations. With multiple product levels, we introduce a new source of demand for float glass products, which need to be produced in campaigns. Demands for secondary products, need to be planned on discrete production lines, and that production is only possible provided a sufficient amount of float glass, which translates into dependent demands. On the other hand, with the introduction of locations, L, we also introduced customer locations as explained previously. With transportation decisions, the model is supposed to reduce all the demands to float level and decide on the campaign plan. Consequently, production schedule, quantities and demand satisfaction plan follow.

This leads us to the idea of projecting demands upstream to float glass, which reduces the network and the problem to parallel machine instance.

Figure 6.2 illustrates a small network consisting of two customer locations *C1* and *C2* and three facilities *L1*, *L2* and *L3*. We assume that transporting products between all of these locations is possible with an associated cost. Circles represent float production lines *F1*, *F2* and *F3*, whereas pentagons represent secondary production lines *S1* and *S2*. Note that, for the sake of simplicity, we do not make distinction between different secondary product types. Let *C1* have demand for float glass. Then there exist three possible projection alternatives, represented with blue solid arrows. Let *C2* have demand for secondary glass. As represented with orange dashed arrows, *L1* and *L2* are two projection alternatives due to *S1* and *S2*. The secondary glass production results in dependent demand for float glass, which can be satisfied from *F1* in *L1* or from *F2* or *F3* in *L3*. We illustrate these second level projection with yellow round dotted arrows.

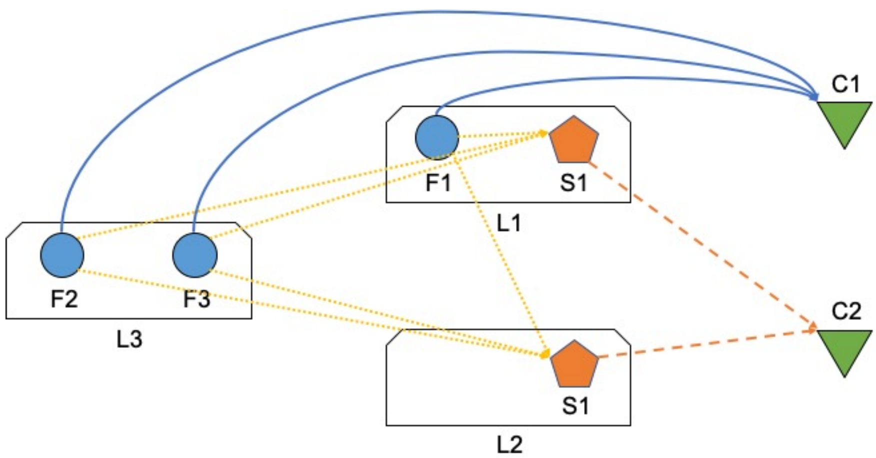

Figure 6.2. Illustration of demand projection alternatives

The cost associated with each projection alternative equals to sum of transportation and production costs for all of the steps included. Note that since we build projection alternatives for all float production lines regardless of their location, it is possible to have multiple paths between the same couple of customer and float line locations, but with different costs unless all included costs sum up to same amount.

Algorithm shown in Figure 6.3 calculates all projection alternatives and associated costs between couples of customer locations and float production locations that is accessible with a projection alternative path.

6.3.2. Algorithm and Variations

In this Section we explain the matheuristics we propose making use of the reduction we explained in Section 6.3.1 of the campaign planning problem. All the algorithms are based on the variation of the Family Transition Based Model FTBMV-MLN, namely Model 6.1.

6.3.2.1. Demand Projection Heuristic. We project demands from customer locations to facilities with production lines using alternatives generated with algorithm in shown Figure 6.3. The algorithm builds the paths from product inventories with demand, down to float inventories with at least one production line. A final alternative a has an associated cost.

Each time algorithm is about to extend the current path, namely create a new alternative a, we increase the associated cost of the alternative by either the *adjusted* unit production or transportation cost. *Adjustment* means multiplying the current cost with parameter $k_{jj'}$ while extending the path with the component product j'.

In demand projection heuristic (DPH), we intuitively project demands to the alternative with the lowest overall cost. This reduces the problem to parallel machine instance. Hence, we can use FTBMV-PM, Model 5.2. Note that this new FTBMV-PM instance also excludes the secondary products and discrete production lines. We recall that FTBMV-PM is a MIP with binary pattern assignment variables δ. In demand projection heuristic, we solve FTBMV-PM with projected demand set D^P. We then solve the original FTBM-MLN with δ variables fixed to the values in the solution of FTBMV-PM denoted as $\bar{\delta}$ in order to calculate the objective function for the original problem. Note that final FTBMV-MLN run is an LP. and Figure 6.4 represents DPH.

GenerateProjectionAlternatives (J, L)

inputs : Set of all products J and locations L with demand

Initialize A as $\emptyset$

foreach inventory (j, l) such that $j \in J, l \in L$ **do**

Create projection alternative a using i

Merge alternatives from Extend(a, j, l) into A

return A

Extend (a, j, l)

inputs : A projection alternative a, an inventory (j, l) to keep extending the alternative a

Initialize A as $\emptyset$

if i has any production alternative r in location l of i **then**

foreach production line r of (j, l) **do**

Create projection alternative a' using a and r

if j has any component j' **then**

Merge alternatives from Extend(a', j', l) into A

else

Add a' to A

else

foreach transportation option $T_{jl'l}$ of j from l' to l **do**

Create projection alternative a' using a and $T_{jl'l}$

Merge alternatives from Extend(a', j, l') into A

return A

Figure 6.3. Generate demand projection alternatives A.

DemandProjectionHeuristic

Project demands with Algorithm 6.3, obtain D^P

Solve *FTBMV-PM* with D^P, obtain $\bar{\delta}$

Solve *FTBMV-MLN* with D and $\delta = \bar{\delta}$

Figure 6.4. Demand Projection Heuristic (DPH).

6.3.2.2. Dynamic Demand Projection Heuristic. In DPH, we project demands to the cheapest projection alternative. Although this is intuitive and preferred, we note that this can result in a costly campaign plan due to capacity restrictions. Demand satisfaction related costs might raise unexpectedly due to aforementioned overflow with not balanced demand allocations. In an optimal solution to an instance with capacity shortage on some production lines, demands would be satisfied from their second, third or even last best projection alternative. Hence, we propose to allow the heuristic to dynamically decide on demand projections. and the number of projection alternatives to be included in the model for each product j will be an input parameter to the heuristic. We represent this number with n.

We introduce new set of variables, ρ_{jlta}, which holds the amount of demand of product j in customer location l in period t to its alternative a. To complete the model, we need to make sure that total amount of each demand is projected properly to allowed alternatives. Eq. (6.38) serves this purpose. Note that $A(j, l, n)$ represent the set of first n demand projection alternatives of product j from location l, ordered by associated cost of the alternative.

$$\sum_{a \in A(j,l,n)} \rho_{jlta} = D_{jlt} \qquad \forall\, j \in J,\ l \in L,\ t \in T \tag{6.38}$$

We create demand satisfaction variables, S, and unsatisfied quantity variables, U, such that for each alternative a, there exists corresponding S and U variables for product j in location l. Also note that we modify Eq.(5.1) such that the right hand side of the equation is equal to the corresponding ρ variable. Note that $A(j)$ represent the set of projection alternatives such that there is an alternative a from product j' and from location l. Following the definition of n, $|A(j)| = n$, Eq.(6.39) is the new demand satisfaction constraints.

$$\sum_{\substack{k \in T \\ k \geq t}} S_{jtk} + U_{jt} = \rho_{j'lta} \quad \forall\, j, j' \in J,\ l \in L,\ a \in A(j),\ t \in T \tag{6.39}$$

Adding Eq. (6.38) to FTBMV-PM and replacing Eq.(5.1) with Eq.(6.39), we obtain Dynamic Demand Projection Model (DDPM). Model 6.5 represents the complete formulation for DDPM.

Model 6.5. *Dynamic Demand Projection Model (DDPM)*

$$\begin{aligned}
\textit{Minimize} \; & \sum_{j\in J}\sum_{t\in T}\left[h_j \; I_{jt} + b_j \; (|T|-t+1) \; U_{jt} + \sum_{\substack{k\in T\\ k\leq t}}(b_j \; (t-k) \; S_{jkt})\right] \\
& + \sum_{r\in R}\sum_{j\in J}\sum_{t\in T} u_{jr}X_{jrt} + \sum_{r\in R}\sum_{p\in P}\sum_{t\in T} c_{pr}\delta_{prt} \\
& + \sum_{r\in R}\sum_{p\in P}\sum_{t\in T} \sum_{\substack{f=f_p^T\\ g=f_s^H\\ (f,g)\notin\Gamma^r(f,g)}} c_{fgr}\theta_{psrt}
\end{aligned}$$

subject to (5.2)

(5.4)–(5.9)

(5.18)–(5.26)

(6.38)

(6.39)

$I_{jt}, U_{jt} \geq 0 \quad \forall(j,t)$

$S_{jtk} \geq 0 \quad \forall(j,t,k\geq t)$

$d^o_{frt} \geq 0 \quad \forall f,r,t$

$X_{jrt} \geq 0 \quad \forall(j,r,t)$

$\rho_{jlta} \geq 0 \quad \forall(j,l,t,a)$

$\delta_{prt} \in \{0,1\} \quad \forall(p,r,t)$

$0 \leq \theta_{fgrt} \leq 1 \quad \forall(f,g,r,t)$

$\gamma^S_{frt}, \gamma^E_{frt} \geq 0 \quad \forall(f,r,t)$

$F_{rt}, B_{rt} \geq 0 \quad \forall(r,t)$

$n^P_{fgrt}, n^S_{fgrt} \geq 0 \quad \forall(f,g,r,t)$

DynamicDemandProjectionHeuristic

inputs : n, number of projection alternatives

Build projection alternatives with Algorithm 6.4, obtain A

Solve *DDPM* with A and n, obtain $\bar{\delta}$

Solve *FTBMV-MLN* with D and $\delta = \bar{\delta}$

Figure 6.5. Dynamic Demand Projection Heuristic (DDPH).

Finally, algorithm shown in Figure 6.5 represent the dynamic demand projection heuristic (DDPH). Note that, the heuristic requires n as in input.

6.3.2.3. Dynamic Demand Projection Heuristic with Relax & Fix. R&F heuristic is an iterative construction heuristic used generally to solve MIP. The main idea in R&F consists of keeping the integrality restriction on a subset of discrete variables which are not yet fixed and relax the remaining unfixed discrete variables. Then solving the resulting sub-MIP, fix subset of the integer variables that are currently part of the integrality subset, and move on to the next iteration. The algorithm stops when all the variables are fixed or the output of the sub-MIP is integral. To construct the R&F algorithm, an important decision is related to the decomposition of the integer variables, in other words deciding on which variables are to be restricted at each iteration. Moreover, determination of the subset to be fixed is another important building block of the R&F. Model 6.6 represent the formulation to solve for a given start S and end E period of integrality restrictions.

Model 6.6. *Dynamic Demand Projection Model with Relax-and-Fix (DDPM-RNF)*

$$\begin{aligned} \mathit{Minimize} \ & \sum_{j\in J}\sum_{t\in T}\left[h_j\ I_{jt} + b_j\ (|T| - t + 1)\ U_{jt} + \sum_{\substack{k\in T\\ k\leq t}}(b_j\ (t-k)\ S_{jkt})\right] \\ & + \sum_{r\in R}\sum_{j\in J}\sum_{t\in T} u_{jr}X_{jrt} + \sum_{r\in R}\sum_{p\in P} c_{pr}\delta_{pr} \\ & + \sum_{r\in R}\sum_{p\in P}\sum_{t\in T}\sum_{\substack{f=f_p^T\\ g=f_s^H\\ (f,g)\notin\Gamma^r(f,g)}} c_{fgr}\theta_{psrt} \end{aligned}$$

subject to (5.2)

(5.4)–(5.9)

(5.18)–(5.26)

(6.38)

(6.39)

$I_{jt}, U_{jt} \geq 0 \quad \forall (j,t)$

$S_{jtk} \geq 0 \quad \forall (j,t,k \geq t)$

$d^o_{frt} \geq 0 \quad \forall f,r,t$

$X_{jrt} \geq 0 \quad \forall (j,r,t)$

$\rho_{jlta} \geq 0 \quad \forall (j,l,t,a)$

$\delta_{prt} \in \{0,1\} \quad \forall (p,r,t \geq S, t \leq E)$

$0 \leq \theta_{fgrt} \leq 1 \quad \forall (f,g,r,t)$

$\gamma^S_{frt}, \gamma^E_{frt} \geq 0 \quad \forall (f,r,t)$

$F_{rt}, B_{rt} \geq 0 \quad \forall (r,t)$

$n^P_{fgrt}, n^S_{fgrt} \geq 0 \quad \forall (f,g,r,t)$

DDPH we described in Section 6.3.2.2 results in a bigger MIP than DPH in terms of number of variables and constraints. Moreover, the campaign planning is a period based problem and it hence is decomposable with respect to periods. We suggest to run the DDPM with R&F with the number of periods to be solved integrality restricted denoted with N^I and the number of periods to be fixed at each iteration denoted with N^F. The heuristic starts from first period and progressively moves the integrality and fixed sets of periods with these parameters. Note that, when a period t is in integrality set, then all the δ variables with index t are defined binary, fixed when in fixed set and relaxed otherwise.

Figure 6.6 illustrates the algorithm based on sub-matrix of the technology matrix including δ variables. Note that we use $m = |P|, n = |R|$ to facilitate the notation.

We illustrate two iterations with starting period being equal to 1 and to 4 with parameters $N^I = 3, N^F = 1$. With red dashed line rectangle we mark the variables with integrality restriction at iterations and with green solid green line we mark the variables to be fixed.

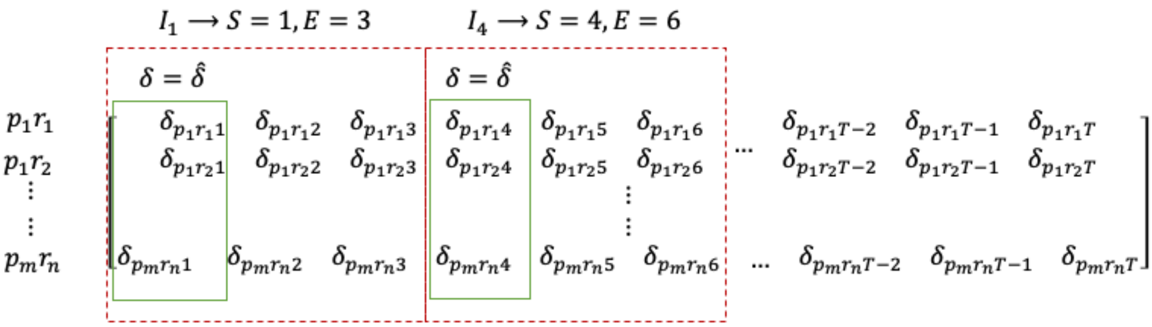

Figure 6.6. DDPH-RNF illustration regarding δ variables.

We describe the dynamic demand projection heuristic with R&F with algorithm shown in Figure 6.7.

DynamicDemandProjectionHeuristicWithRNF

inputs : Number of periods to be set integer N^I, number of periods to be fixed at each iteration N^F, where $N^I \geq N^F$

Initialize current starting period S with 0 and current ending period E with N^I

do

Solve *DDPM-RNF* with S and E

Fix δ_{prt} with $S \leq t \leq S + N^F$

$S \leftarrow S + N^F$

$E \leftarrow S + N^I$

while $E < |P|$;

Figure 6.7. Dynamic Demand Projection Heuristic with Relax-and-Fix (DDPH-RNF).

7. COMPUTATIONAL RESULTS

In this Chapter, we present the numerical results for all the formulations and methods. Following problem structure, we present the results for Single Machine instance in Section 7.1, for Parallel Machine instance in Section 7.2.4 and finally for Multiple Level Network in Section 7.3. We implemented all the formulations and algorithms with C# language of the .NET Framework and used commercial solver CPLEX (12.8). We also used Gurobi (8.1) for comparing commercial solvers in Section 7.1. We executed all experiments on a PC with Intel Core i7-8750H CPU 2.20 GHz and 32 GB RAM.

7.1. Single Machine

7.1.1. Dataset and Problem Instances

The data used in the numerical experiments is based on real life data provided by a major float glass manufacturer in Turkey. Hence, the data is realistic in terms of production, setup and cost perspective. The data set contains 153 unique products of different color, size, quality, coating, thickness and packaging type attributes.

Color is the primary attribute affecting the duration and the cost of a changeover. Hence, we include color in the family structure. In addition, coating is another attribute that requires setup between products of the same color. Hence, color and coating will be considered as attributes that form a family. Moreover, in order to investigate the significance of adding or removing an attribute in family structure, we will work with three different structures. We can enumerate them as follows:

- **Color:** The simplest structure. Only color forms a family, and all coating types are considered in the same family

- **Color & C/NC:** In addition to color, coating is incorporated into family structure in a binary form: **C** = Coated, **NC** = Not Coated
- **Color & Coating:** Both color and coating attributes are considered in families.

There are three colors, namely fume (FM), bronze (BR) and blue (MV), and three coating types, namely without coating (Z), pyrolitic (P) and titanium (T). For each different family structure explained above, we have 3, 6 and 8 families respectively aggregating 153 unique products.

7.1.2. Single Machine Numerical Results

In order to compare the performances of the four models proposed with the data set explained in Section 7.1.1 we designed a set of run instances. We can list the main attributes for the instances as follows:

- **Number of Periods:** 4, 6 and 8 periods
- **Formulation:** PTBM, FTBM, PTBMV and FTMBV
- **Family Structure:** Color, Color & C/NC and Color & Coating

Table 7.1 shows values for the number of continuous and binary variables and the number of constraints. The mfamily structures Color, Color & C/NC and Color & Coating have 24, 115 and 135 patterns respectively.

We note that the number of patterns depends on the family structure. Similarly, the number of variables in each formulation depends on the formulation and the number of periods in addition to the family structure. The number of binary variables, on the other hand, depends on the number of patterns and periods (δ_{pt}).

We can observe that the number of variables and constraints increase in all formulations with respect to the family structure. However, the increase rate is much higher in Pattern Transition Based (PTB) models (PTBM and PTBMV).

Table 7.1. Characteristics of run instances.

		Model			
Structure	**Number of**	**FTBM**	**FTBMV**	**PTBM**	**PTBMV**
Col.	Binary Vars.	96	96	96	96
	Continuous Vars.	2426	2426	4187	4187
	Constraints	1312	1315	9981	9984
Col. & C/NC	Binary Vars.	460	460	460	460
	Continuous Vars.	2621	2621	42716	42716
	Constraints	2412	2361	185886	170670
Col. & Coat.	Binary Vars.	540	540	540	540
	Continuous Vars.	2769	2769	57860	57860
	Constraints	3000	2877	245986	215494
Col.	Binary Vars.	144	144	144	144
	Continuous Vars.	4125	4125	7064	7064
	Constraints	1997	2002	16434	16439
Col. & C/NC	Binary Vars.	690	690	690	690
	Continuous Vars.	4440	4440	71273	71273
	Constraints	3701	3616	309423	284063
Col. & Coat.	Binary Vars.	810	810	810	810
	Continuous Vars.	4680	4680	96509	96509
	Constraints	4639	4434	409537	358717
Col.	Binary Vars.	192	192	192	192
	Continuous Vars.	6205	6205	10322	10322
	Constraints	2703	2710	22908	22915
Col. & C/NC	Binary Vars.	920	920	920	920
	Continuous Vars.	6640	6640	100211	100211
	Constraints	5011	4892	432981	397477
Col. & Coat.	Binary Vars.	1080	1080	1080	1080
	Continuous Vars.	6972	6972	135539	135539
	Constraints	6299	6012	573109	501961

The number of variables and constraints are expected to be much higher in Pattern Based models. This is the case for family structure Color & Coating and eight periods instance. However, we observe that when coating is not selected as a family-forming attribute the results are somewhat surprising. For instance when we compare PTBM and PTBMV in Color family structure and four periods instance, we see that the number of variables remains constant and that the number of constraints increases in Variant version. We observe that the reason behind such a case is the following: once the coating attribute is removed from the family structure, the family sequence setup restrictions disappear as it is possible to change colors in any sequence (with different setup durations). Hence, PTBM does no contain Eq. (4.18) and its variant version contains Eq. (4.39). A similar situation is also observed in Family Transition Based models.

In order to analyse the efficiency of the pattern preprocessing, let us share the details about the number of patterns per family structure. In Color structure, algorithm in Figure 4.3 generates 42 patterns and algorithm shown in Figure 4.4 eliminates 18 of them resulting in 43% decrease. Similarly, respective numbers for Color & C/NC are 165, 115 and 30%, and for Color & Coating are 171, 135 and 21%. Note that the number of patterns decreases by 31% on average, which is important in terms of performance since the number of binary variables depends on the number of patterns.

Regarding the solution performance, let us first observe the Linear Programming (LP) relaxation objective values of the formulations. Table 7.2 shows the objective values of LP relaxation of the proposed formulations. We observe that Family Transition Based (FTB) formulations generate significantly tighter LP relaxation objectives compared to PTB models.

Moreover, for both PTB and FTB models, variant formulations produce higher LP relaxation objectives in all run instances compared to their respective original formulations, which is in alignment with Propositions 4.2 and 4.3.

Table 7.2. LP relaxation objective values.

	Number of		LP-Relaxation Objective of			
Structure	**Periods**	**Patterns**	**PTBM**	**PTBMV**	**FTBM**	**FTBMV**
Col.	4	24	2103	3123	100603	100603
Col. & C/NC	6	24	18498	19411	232606	232606
Col. & Coat.	8	24	41707	44333	436680	436680
Col.	4	115	4438.83	5609	100608	100608
Col. & C/NC	6	115	20401	21971	232611	232611
Col. & Coat.	8	115	48904	52107	436686	436686
Col.	4	135	5296.56	6478	100656	100656
Col. & C/NC	6	135	22239	23881	232711	232721
Col. & Coat.	8	135	51904	55198	436847	436857

We implemented a general purpose optimization layer in our implementation. It enables us to use both CPLEX and Gurobi solvers.

Table 7.3 and Table 7.4 illustrate Central Processing Unit (CPU) time in seconds, relative MILP gap and incumbent solution objective value per run instance for CPLEX and Gurobi respectively. All instances are solved with a time limit of 8 hours (28800 seconds).

We note that for each family structure and number of period combination, at least one of the formulations was able to find an optimal solution. Moreover, some of the solution runs, such as PTBM in eight periods and Color & C/NC family structure, were able to find an optimal objective value but were not able to prove the optimality. Regarding the formulations, we note that in all instances FTB models outperform PTB models. We investigate the performances of CPLEX for the sake of simplicity in summary.

Considering FTBM and its variant, FTBMV, the variant performs better than the original formulation regarding computational time except a single instance, 6 periods and Color as family structure. We observe that FTBM finds an optimal solution in the root node, whereas FTBMV also finds an optimal solution at the root but couldn't prove the optimality without exploring 383 nodes consting an extra second.

On the other hand, PTBMV consistently performs worse than PTBM regarding computational time. To further investigate, we checked the solver logs and observed that root node solution time is consistently taking much longer in variant formulations. For example, in 8 periods and Color & Coating family structure, root node processing takes 1708 seconds in PTBMV while 201 seconds in PTBM. A potential reason for such a difference is related to PTBM having many more constraints than its variant except for one case explained above. PTBM has more and sparser constraints as in Eq. (4.18) whereas the variant PTBMV has less and denser set of constraints with Eq. (4.39). Considering the solvers' working mechanism of working with sparse algebra, we can explain the difference in computational performance.

Table 7.3. CPLEX performance by run instance.

Model	**Structure**	**Nb. of Periods**	**Nodes**	**CPU Time(s)**	**Relative Gap**	**Objective**
FTBM	Col.	4	0	0	0%	2877988.95
FTBMV	Col.	4	0	0	0%	2877988.95
PTBM	Col.	4	0	2	0%	2877988.95
PTBMV	Col.	4	204	5	0%	2877988.95
FTBM	Col. & C/NC	4	1817	8	0%	3809582.14
FTBMV	Col. & C/NC	4	1500	8	0%	3809582.14
PTBM	Col. & C/NC	4	2720	118	0%	3809582.14
PTBMV	Col. & C/NC	4	2429	823	0%	3809582.14
FTBM	Col. & Coat.	4	3023	18	0%	5022123.81
FTBMV	Col. & Coat.	4	1574	14	0%	5022123.81
PTBM	Col. & Coat.	4	2863	161	0%	5022123.81

Table 7.3. CPLEX performance by run instance. (cont.)

PTBMV	Col. & Coat.	4	2358	1282	0%	5022123.81
FTBM	Col.	6	0	1	0%	3120228.39
FTBMV	Col.	6	383	2	0%	3120228.39
PTBM	Col.	6	1513	15	0%	3120228.39
PTBMV	Col.	6	1517	37	0%	3120228.39
FTBM	Col. & C/NC	6	16945	223	0%	4063448.72
FTBMV	Col. & C/NC	6	22796	184	0%	4063448.72
PTBM	Col. & C/NC	6	62674	11514	0%	4063448.72
PTBMV	Col. & C/NC	6	17324	28800	37%	4125956.22
FTBM	Col. & Coat.	6	43557	813	0%	4968043.49
FTBMV	Col. & Coat.	6	23491	426	0%	4968043.49
PTBM	Col. & Coat.	6	69062	16175	0%	4968043.49
PTBMV	Col. & Coat.	6	10976	28800	51%	5242480.08
FTBM	Col.	8	3339	16	0%	3710685.29
FTBMV	Col.	8	2699	11	0%	3710685.29
PTBM	Col.	8	7144	91	0%	3710685.29
PTBMV	Col.	8	8722	399	0%	3710685.29
FTBM	Col. & C/NC	8	195853	3433	0%	4316352.78
FTBMV	Col. & C/NC	8	84820	1901	0%	4316352.78
PTBM	Col. & C/NC	8	53867	28800	38%	4316352.78
PTBMV	Col. & C/NC	8	9053	28800	51%	4316352.78
FTBM	Col. & Coat.	8	269178	15487	0%	5015743.52
FTBMV	Col. & Coat.	8	257728	8873	0%	5015743.52
PTBM	Col. & Coat.	8	51628	28800	26%	5015743.52
PTBMV	Col. & Coat.	8	6103	28800	62%	5737056.49

Table 7.4. Gurobi performance by run instance.

Model	**Structure**	**Nb. of Periods**	**Nodes**	**CPU Time(s)**	**Relative Gap**	**Objective**
FTBM	Col.	4	136	0	0%	2877988.95
FTBMV	Col.	4	148	0	0%	2877988.95
PTBM	Col.	4	622	3	0%	2877988.95
PTBMV	Col.	4	888	5	0%	2877988.95
FTBM	Col. & C/NC	4	4818	6	0%	3809582.14
FTBMV	Col. & C/NC	4	6753	8	0%	3809582.14
PTBM	Col. & C/NC	4	3552	423	0%	3809582.14
PTBMV	Col. & C/NC	4	3753	1585	0%	3809582.14
FTBM	Col. & Coat.	4	3552	16	0%	5022123.81
FTBMV	Col. & Coat.	4	4456	16	0%	5022123.81
PTBM	Col. & Coat.	4	3807	459	0%	5022123.81
PTBMV	Col. & Coat.	4	6232	5155	0%	5022123.08
FTBM	Col.	6	733	1	0%	3120228.39
FTBMV	Col.	6	461	1	0%	3120228.39
PTBM	Col.	6	1607	17	0%	3120228.39
PTBMV	Col.	6	1618	23	0%	3120228.39
FTBM	Col. & C/NC	6	33341	79	0%	4063448.72
FTBMV	Col. & C/NC	6	25937	58	0%	4063448.72
PTBM	Col. & C/NC	6	32195	4326	0%	4063448.72
PTBMV	Col. & C/NC	6	1509	28800	64%	4443022.34
FTBM	Col. & Coat.	6	45772	175	0%	4968043.49
FTBMV	Col. & Coat.	6	55325	202	0%	4968043.49
PTBM	Col. & Coat.	6	39070	10074	0%	4968043.49
PTBMV	Col. & Coat.	6	4302	28800	64%	5074585.78
FTBM	Col.	8	2210	7	0%	3710685.29
FTBMV	Col.	8	1236	3	0%	3710685.29
PTBM	Col.	8	2831	45	0%	3710685.29

Table 7.3. Gurobi performance by run instance. (cont.)

PTBMV	Col.	8	3235	70	0%	3710685.29
FTBM	Col. & C/NC	8	146623	1083	0%	4316352.78
FTBMV	Col. & C/NC	8	153040	1745	0%	4316352.78
PTBM	Col. & C/NC	8	71633	28800	21%	4316352.66
PTBMV	Col. & C/NC	8	1535	28800	69%	4316352.78
FTBM	Col. & Coat.	8	176128	16785	0%	5015743.52
FTBMV	Col. & Coat.	8	135685	17534	0%	5015743.52
PTBM	Col. & Coat.	8	25495	28800	30%	5015743.52
PTBMV	Col. & Coat.	8	1272	28800	72%	5496831.64

A solver outperforms the other if it obtains a solution with lower optimality gap. If both obtain an optimal solution within the time limit, then whichever proves optimality earlier is noted as the winner. Let us summarize the number of "wins" per solver as follows:

- **4 Periods:** Gurobi wins 5 times while CPLEX wins remaining 7
- **6 Periods:** Gurobi wins 8 times while CPLEX wins remaining 4
- **8 Periods:** Gurobi wins 7 times while CPLEX wins remaining 5

We observe that, in more cases Gurobi outperforms CPLEX and especially in FTB models, Gurobi obtains provably optimal solutions faster than CPLEX. As the problem instance becomes more complex, Gurobi tends to outperform CPLEX. However, in the most complex instance, which is 8 periods with Color & Coating family structure, CPLEX finds a provably optimal solution in 8873 seconds whereas Gurobi is able to solve the instance in 17534, which is almost twice the time. Moreover, in smaller instances, those with 4 periods namely, CPLEX outperforms Gurobi.

Hence, we can conclude that there is no clear superiority of one solver to the other. Nevertheless, we will use FTBMV and Gurobi for further experiments, being the combination most frequently performing better than the others.

7.1.3. Business Insights

Analysis presented in Section 7.1.2 discusses the problem and formulations in detail from a mathematical point of view. Set of experiments up to now measure the performance of different formulations proposed. However, since the problem has some unique challenges, it is also valuable to elaborate the analysis on some business insights perspective. Our main goal is to observe the characteristics of the generated campaign plans with respect to different business scenarios.

As stated in Section 7.1.2, we will use FTBMV in a set of experiments for testing further scenarios. Our main goal in the next is to analyse the changes in number campaigns and average duration per campaign overall. Total setup duration driven by campaign plan is also another metric to be observed. We expect to gather further insights from other business indicators such as average total ending inventory per month and total backlogged or unsatisfied demand.

Costs associated with inventory holding and demand backlog/unsatisfaction are subject to some business requirements and assumptions. Moreover, setup costs have a crucial role in campaign decisions being a significant expense item and having physical counterpart. Since all these costs mentioned are in the objective function to be minimized, we decided to design a new set of run instances that will enable us to observe the marginal effect of each cost component to the resulting campaign plan.

We adapt an approach similar to [6] in order to evaluate effects of cost components. We first assume a baseline run instance with family structure Color & Coating and 8 periods. Then, for each cost component, we solve the campaign planning problem having corresponding coefficients multiplied with 0.1, 0.2, 0.5, 2, 5 and 10.

In each case, we observe the changes in various measures such as the number of campaigns, total setup duration and average ending inventory. Figure 7.1 shows an optimal campaign plan for our baseline instance.

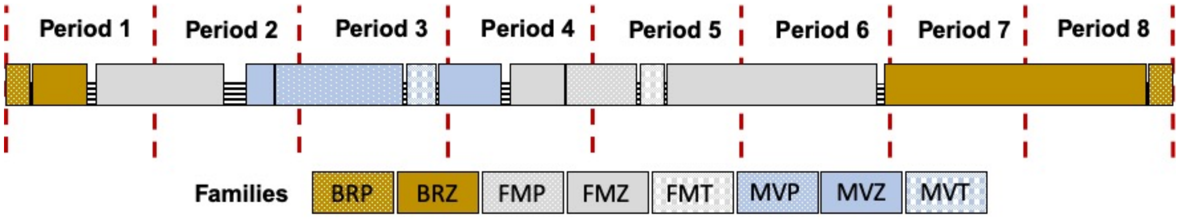

Figure 7.1. Optimal Campaign Plan for Baseline Instance

We first analyze the effect of setup costs. Figure 7.2 shows some metrics that will help us interpret the behavior of the outcoming campaign plans compared to the expectations. In each one of the charts, term Mx corresponds to a run instance where M stands for the multiplier used. Note that 1x is the Baseline instance. With increasing the setup costs, we expect to have fewer setups, which is validated with Figure 7.2 (a). Considering the average campaign duration, although the trend is increasing as expected with fewer campaigns per family, in 5x instance we observe the measure against our expectation. The difference is driven by family BRP, which in 5x instance has a single campaign of 5.06 days whereas in 2x instance there are two BRP campaigns with average duration of 17.72 days. We further observe that the ending inventory at the end of the planning horizon for family BRP is 14247 in 2x instance whereas this figure is only 331 in 5x instance. The inventory to be held shifted to FMZ family in 5x instance, which did not have any ending inventory in 2x instance. We anticipate that with increased setup costs, model could decrease the overall costs with such a combination regarding inventory holding costs. With fewer number of campaigns, the total setup duration spent is expected to be less as well, which can be observed in Figure 7.2 (c). With longer campaign durations higher amount of inventory is expected to be carried, which we validate with Figure 7.2 (b) and we observe a similar behavior for total backlogged and unsatisfied demand quantity.

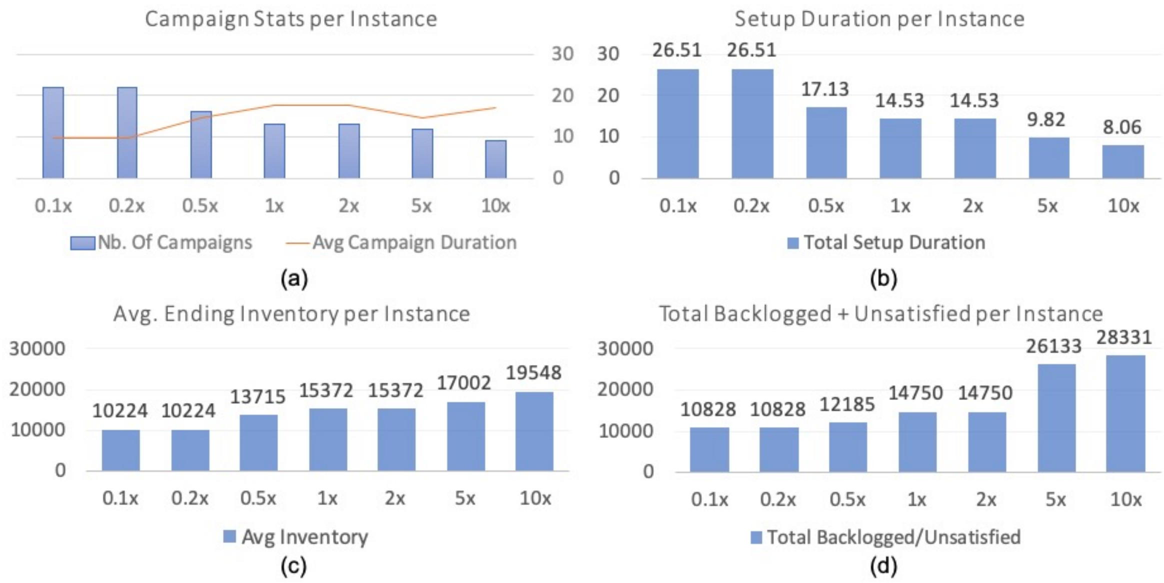

Figure 7.2. Measures for instances with modified setup coefficients

Figure 7.3 shows the effects of the changes in backlog coefficients. With increasing backlog costs, in order to decrease the cost due to backlogging, we expect to have more campaigns in shorter duration. Figures 7.3 (a) and (b) illustrate the increase in both number of campaigns and total setup duration. However, average campaign duration fluctuates even though the trend is downwards.

Clearly, with increasing backlog cost, models tend to have less and less backlogged demand and average ending inventory is also decreasing since there is a larger number of shorter campaigns.

Inventory holding cost is the expense item with the least effect on resulting campaign plans as observed in Figure 7.4. With increasing inventory cost, we expect to have more campaigns having shorter duration to avoid holding more inventory longer. This is observed with Figure 7.4 (a). Also with more campaigns, we observe eventually longer total setup duration. The average ending inventory tends to decrease but only a significant change in inventory costs can drive this.

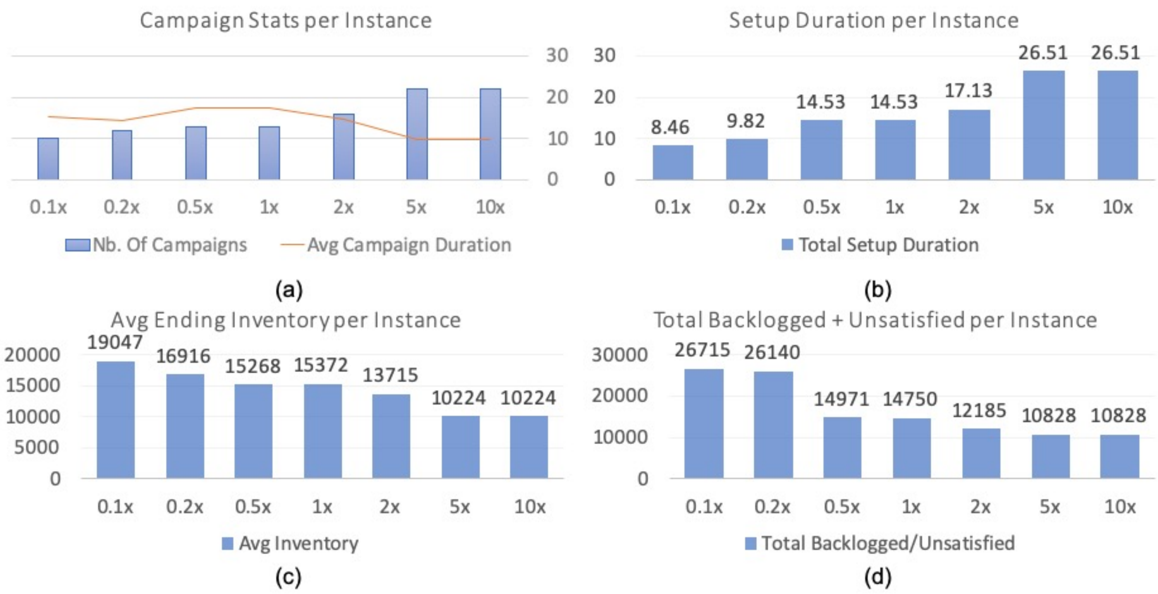

Figure 7.3. Measures for instances with modified backlog coefficients

Figure 7.4. Measures for instances with modified inventory coefficients

7.2. Parallel Machine

7.2.1. Preliminary Tests on Family and Pattern Transition Based Model Extensions

In this Section, we present the results of preliminary tests on formulations defined in Sections 5.2 and 5.3. Similar to the single machine instance, we use a dataset corresponding to a realistic data set. As we focus on parallel machine environment in this Section, we extracted a dataset from the entire production network of the company, including identical and unrelated parallel machines. Moreover, we run the experiments with 5 different data sets including 8 periods of planning horizon, and with color and coating in family structure. Table 7.6 shows the number of patterns, the number of both continuous and binary variables and the number of constraints. We observe that the number of continuous variables and constraints are much higher in PTBMV compared to FTBMV. In detail, PTBMV has 12 times more continuous variables and 19 times more constraints thant FTBMV. Moreover, we note that machines being identical or unrelated does not have direct impact on the number of patterns.

Regarding the solution performance, we present the results in 7.6 Let us first focus on the Linear Programming (LP) relaxation objectives. We observe that FTBMV-PM generate better LP relaxation objectives compared to PTBMV-PM 8% on average for unrelated machines and more than 20% on average for identical machines. We note that in these preliminary results, none of the formulations were able to find an optimal solution within 8 hour time limit. In all instances, FTBMV-PM models outperform PTBMV-PM in all instances. In particular, FTBMV-PM finds solutions with better relative gaps than FTBM. Another note is that, PTBMV-PM can explore much less nodes in the search tree compared to FTBMV-PM. Preliminary results reveal that both PTBMV-PM and FTBMV-PM are unable to solve the problems to optimality within 8-hours. For the rest, we will focus on building a nouvel approach to solve the problem with better quality and in less time. Moreover, since PTBMV-PM proves to be inefficient for larger problems, we will proceed with FTBMV-PM only.

Table 7.5. Characteristics of run instances.

	Number of			Formulation	
Structure	**Resources**	**Patterns**	**Number of**	**PTBMV**	**FTBMV**
Identical	2	170	Bin.	1360	1360
			Cont.	111020	9286
			Constr.	414794	21356
Unrelated	2	184	Bin.	1472	1472
			Cont.	132521	12465
			Constr.	469788	24736
Identical	3	255	Bin.	2040	2040
			Cont.	164201	11600
			Constr.	621257	31100
Unrelated	3	252	Bin.	2016	2016
			Cont.	167673	15018
			Constr.	600917	32633

Table 7.6. Characteristics of run instances.

Resource						
Structure	**Count**	**Model**	**LP-Relaxation**	**Nodes**	**CPU Time(s)**	**Relative Gap**
Identical	2	PTBMV-PM	101985763.6	1	28800	23.56%
		FTBMV-PM	129336184.8	113783	28800	1.20%
Unrelated	2	PTBMV-PM	122689627.1	89	28800	14.22%
		FTBMV-PM	133218387.9	54895	28800	1.75%
Identical	3	PTBMV-PM	152978645.4	1	28800	24.90%
		FTBMV-PM	194004277.2	50707	28800	1.10%
Unrelated	3	PTBMV-PM	183072382.1	1	28800	13.13%
		FTBMV-PM	199918218.2	38975	28800	1.14%

7.2.2. Data Set and Problem Instances

The data used in the numerical experiments, similar to single machine instance, is based on real life data from the same major float glass manufacturer in Turkey. Hence, the data is realistic in terms of production, setup and cost perspective. We defined three different family structures in Section 7.1.1, which have effect on the complexity of the problem. Having color and coating in the structure is the most complex case, and we will use this structure in our experiments.

The difficulty in problem stems from the decision on pattern allocation to resources. In FTBMV-PM, we need to make this decision for each resource and macro period. In B&P for CPM-EP, we progressively generate the extended patterns. The number resources and the number of macro periods effect the number of all possible extended patterns. Hence, it is important to test the performance of B&P for CPM-EP and FTBMV-PM with datasets ranging in terms of number of resources and periods.

We have two base datasets each containing 1482 unique products of different color, size, quality, coating, thickness and packaging type attributes. Two datasets differ from each other with respect to the number of machines; one has 3 production lines whereas the other has 5 lines. In each case, all the machines are unrelated in terms of production speed for products. For each of these base datasets, we have 5 different demand scenarios, and for each one of the demand scenarios, we solve the problem for 5 different planning horizon length. In detail, we solve the instances for 4, 6, 8, 10 and 12 periods.

We assume the micro-period length as days and note that it can be modified according to data or practical needs. Moreover, we adjusted the minimum production durations and setup times to be allocated over period boundaries between adjacent patterns in FTBMV-PM.

7.2.3. Preliminary Tests on Node Selection Strategy

We define different node selection strategies, namely DFS, BFS and BBS, in Section 5.4.5.2 to be employed throughout the solution. We first run some preliminary tests to determine whether a strategy outperforms the others. In order to capture performance against different complexities, we randomly selected 3 instances for combinations of 3 and 5 machines with 4, 8 and 12 periods. We limit the overall solution time to 3600 seconds for all instances. Note that we do not employ the adaptive root node processing explained in Section 5.4.5.4 in our preliminary tests.

Tables 7.7 and 7.8 show the outputs of preliminary test results with 3 and 5 parallel machines respectively. Note that *Run* column is a representation for the run instance with the example that "m:3-p:4-d:3" stand for instances with 3 parallel machines (m), 4 macro periods (p) and demand scenario (d) 3. For each test instance and node selection strategy, we report the MIP optimality gap ("Gap" column) and the number of nodes explored in the search tree ("Nodes" column).

Results show that on 3 machine tests, BF outperforms the other strategies in 7 out of 9 runs. DF and BB strategies each are able to provide the best outputs in 1 instance. On 5 machine tests on the other hand, the outlook is exactly on the contrary. DF outperforms other methods in 7 out of 9 runs with BB and BF having 1 best performance each. First conclusion is that regarding the node selection strategy in the solution approach we propose, BB is outperformed by the other methods. Moreover, for different machine configurations we observe significantly better performing strategies. Finally, in all test instances BB has the least number of nodes process from the search tree. Hence, for the remainder of our numerical experiments, we will employ BF in 3 machine instances and DF in 5 machine instances.

Table 7.7. Test results on node selection strategy on 3 machines.

	BB		DF		BF	
Run	**Gap**	**Nodes**	**Gap**	**Nodes**	**Gap**	**Nodes**
m:3-p:4-d:1	11.26%	**9052**	11.26%	8932	**9.29%**	5216
m:3-p:4-d:3	10.58%	**8840**	10.58%	8438	**9.31%**	3733
m:3-p:4-d:5	**10.52%**	**9900**	10.86%	9454	11.56%	5341
m:3-p:4-avg	10.79%	**9264**	10.90%	8941	**10.05%**	4763
m:3-p:8-d:2	14.69%	**4485**	14.69%	4247	**13.21%**	137
m:3-p:8-d:3	13.32%	1139	13.32%	**1184**	**13.31%**	151
m:3-p:8-d:4	15.43%	**1163**	15.43%	1106	**11.66%**	132
m:3-p:8-avg	14.48%	**2262**	14.48%	2179	**12.73%**	140
m:3-p:12-d:2	14.15%	11	14.15%	**21**	**14.07%**	11
m:3-p:12-d:3	13.90%	17	13.90%	**22**	**13.20%**	6
m:3-p:12-d:5	14.36%	6	**13.45%**	**8**	14.36%	1
m:3-p:12-avg	14.14%	11	**13.83%**	**17**	13.88%	6

Table 7.8. Test results on node selection strategy on 5 machines.

	BB		DF		BF	
Run	**Gap**	**Nodes**	**Gap**	**Nodes**	**Gap**	**Nodes**
m:5-p:4-d:1	5.16%	6766	4.31%	**9331**	**3.67%**	1404
m:5-p:4-d:4	10.16%	505	**2.89%**	**8639**	5.84%	1545
m:5-p:4-d:5	5.63%	410	**4.05%**	**8408**	6.38%	2148
m:5-p:4-avg	6.98%	2560	3.75%	**8793**	**5.30%**	1699
m:5-p:8-d:1	6.40%	286	**6.32%**	**1000**	6.57%	157
m:5-p:8-d:2	7.41%	10	**5.39%**	**252**	7.09%	134
m:5-p:8-d:4	7.57%	14	**3.87%**	**364**	6.58%	147
m:5-p:8-avg	7.13%	103	5.19%	**539**	**6.75%**	146
m:5-p:12-d:1	10.04%	1	**10.04%**	1	10.04%	1
m:5-p:12-d:4	13.61%	1	**13.61%**	1	13.61%	1
m:5-p:12-d:5	**11.86%**	1	12.00%	1	12.00%	1
m:5-p:12-avg	11.84%	1	11.88%	1	**11.88%**	1

7.2.4. Parallel machine numerical results

Considering dataset and algorithm settings described in Section 7.2.2, our numerical experiments consist of 100 instances for the B&P algorithm. For the comparison, we take FTBMV-PM runs for all datasets, which count up to additional 50 instances. Similar to our approach in preliminary tests explained in Section 7.2.3, we limit the overall solution time to 1-hour for all instances. For each unique dataset instance, we also solve the model with FTBMV-PM with a 1-hour time limit. Moreover, since our B&P algorithm is implemented to work on single thread, to have a fair comparison we limit the number of threads to be used by CPLEX as one.

Tables 7.9 and 7.10 show the outputs of B&P for CPM-EP and FTMBV-PM for 3 and 5 parallel machines respectively. We report the average MIP optimality gap from all run instances for FTBMV-PM, B&P and Adaptive B&P. Note that *Run* column is a smart representation indicating number of machines with m and number of periods with p as explained in previous Section. Note that, with the increase in number of periods, the algorithms obtain solution with higher optimality gaps. Overall, FTBMV-PM outperforms both of the B&P approaches for 3 machine instances, whereas on 5 machine instances, B&P algorithms significantly outperform FTBMV-PM. Amongst B&P algorithms proposed, classical approach performs better than adaptive in 3 out of the 5 3 machine instances. On the other hand, in 5 machine instances, adaptive approach outperforms classical approach in 4 out of 5.

Table 7.9. Average gap FTBMV-PM compared to B&P algorithms on 3 machines.

Run	**FTBMV**	**B&P**	**B&P Adaptive**
m:3-p:4	**0.00%**	10.33%	10.80%
m:3-p:6	**2.30%**	10.26%	10.26%
m:3-p:8	**4.52%**	12.91%	11.71%
m:3-p:10	**9.72%**	13.28%	12.87%
m:3-p:12	**10.50%**	14.11%	13.22%

Table 7.10. Average gap FTBMV-PM compared to B&P algorithms on 5 machines.

Run	FTBMV	B&P	B&P Adaptive
m:5-p:4	6.43%	4.09%	**4.08%**
m:5-p:6	68.26%	5.85%	**5.63%**
m:5-p:8	81.59%	**5.81%**	6.76%
m:5-p:10	87.35%	7.20%	**7.10%**
m:5-p:12	88.68%	11.62%	**8.42%**

Tables 7.11 and 7.12 present explicit results for all run instances comparing FTBMV-PM with B&P with classical root node processing B&P with adaptive root node processing for 3 parallel machines and for 5 parallel machines respectively. Note that, with our approach, we need to have finished the root node processing in order to provide an optimality gap. In cases where the B&P algorithms require more than the given time limit to provide an optimality gap, we note the duration as *'time'*.

Table 7.11. Models compared on 3 machines.

	FTBMV-PM		**B&P**		**B&P Adaptive**	
Run	**Gap**	**Time**	**Gap**	**Time**	**Gap**	**Time**
m:3-p:4-d:1	**0.00%**	469	9.29%	3600	12.25%	3600
m:3-p:4-d:2	**0.00%**	340	10.02%	3600	10.02%	3600
m:3-p:4-d:3	**0.00%**	604	9.31%	3600	9.31%	3600
m:3-p:4-d:4	**0.00%**	370	11.49%	3600	11.44%	3600
m:3-p:4-d:5	**0.00%**	287	11.56%	3600	10.96%	3600
m:3-p:6-d:1	**2.19%**	3600	10.04%	3600	10.04%	3600
m:3-p:6-d:2	**1.78%**	3600	11.74%	3600	11.74%	3600
m:3-p:6-d:3	**2.54%**	3600	10.08%	3600	10.08%	3600
m:3-p:6-d:4	**3.18%**	3600	8.62%	3600	8.62%	3600
m:3-p:6-d:5	**1.80%**	3600	10.84%	3600	10.84%	3600
m:3-p:8-d:1	**4.12%**	3600	14.32%	3600	11.21%	3600
m:3-p:8-d:2	**3.29%**	3600	13.21%	3600	11.96%	3600

Table 7.11. Models compared on 3 machines. (cont.)

m:3-p:8-d:3	**4.52%**	3600	13.31%	3600	12.14%	3600
m:3-p:8-d:4	**4.76%**	3600	11.66%	3600	11.41%	3600
m:3-p:8-d:5	**5.93%**	3600	12.07%	3600	11.83%	3600
m:3-p:10-d:1	**7.17%**	3600	12.58%	3600	13.78%	time
m:3-p:10-d:2	**11.43%**	3600	13.64%	3600	12.20%	time
m:3-p:10-d:3	**10.80%**	3600	15.39%	3600	14.29%	time
m:3-p:10-d:4	**8.48%**	3600	12.61%	3600	11.46%	time
m:3-p:10-d:5	**10.71%**	3600	12.16%	3600	12.60%	time
m:3-p:12-d:1	13.62%	3600	13.10%	3600	**12.54%**	time
m:3-p:12-d:2	**10.06%**	3600	14.07%	time	12.69%	time
m:3-p:12-d:3	**9.81%**	3600	13.20%	3600	12.94%	time
m:3-p:12-d:4	**10.82%**	3600	15.84%	time	12.89%	time
m:3-p:12-d:5	**8.21%**	3600	14.36%	time	15.03%	time

Table 7.12. Models compared on 5 machines.

	FTBMV-PM		**B&P**		**B&P Adaptive**	
Run	**Gap**	**Time**	**Gap**	**Time**	**Gap**	**Time**
m:5-p:4-d:1	5.21%	3600	4.31%	3600	**4.31%**	3600
m:5-p:4-d:2	**2.87%**	3600	5.43%	3600	5.35%	3600
m:5-p:4-d:3	7.94%	3600	3.75%	3600	**3.75%**	3600
m:5-p:4-d:4	10.86%	3600	**2.89%**	3600	2.95%	3600
m:5-p:4-d:5	5.29%	3600	4.05%	3600	**4.05%**	3600
m:5-p:6-d:1	75.09%	3600	6.17%	3600	**6.17%**	3600
m:5-p:6-d:2	83.09%	3600	5.07%	3600	**5.07%**	3600
m:5-p:6-d:3	77.27%	3600	6.66%	3600	**5.58%**	3600
m:5-p:6-d:4	77.29%	3600	5.90%	3600	**5.89%**	3600

Table 7.12. Models compared on 5 machines. (cont.)

m:5-p:6-d:5	28.56%	3600	5.44%	3600	**5.44%**	3600
m:5-p:8-d:1	82.05%	3600	6.32%	3600	**6.29%**	3600
m:5-p:8-d:2	83.02%	3600	**5.39%**	3600	6.32%	3600
m:5-p:8-d:3	82.02%	3600	**7.00%**	3600	8.17%	3600
m:5-p:8-d:4	79.06%	3600	**3.87%**	3600	5.62%	3600
m:5-p:8-d:5	81.80%	3600	**6.48%**	3600	7.39%	3600
m:5-p:10-d:1	87.36%	3600	**7.31%**	3600	7.40%	time
m:5-p:10-d:2	87.38%	3600	7.01%	3600	**5.95%**	time
m:5-p:10-d:3	87.34%	3600	7.42%	3600	**7.00%**	time
m:5-p:10-d:4	87.31%	3600	**6.72%**	3600	7.06%	time
m:5-p:10-d:5	87.37%	3600	**7.54%**	3600	8.08%	time
m:5-p:12-d:1	88.66%	3600	10.04%	time	**8.04%**	time
m:5-p:12-d:2	88.69%	3600	11.66%	time	**9.29%**	time
m:5-p:12-d:3	88.65%	3600	10.80%	time	**8.43%**	time
m:5-p:12-d:4	88.63%	3600	13.61%	time	**8.45%**	time
m:5-p:12-d:5	88.77%	3600	12.00%	time	**7.87%**	time

We compare the results similar to the approach described in Section 7.1.2. An algorithm outperforms the other if it obtains a solution with lower optimality gap. As per the aggregated results, in 3 machine instances FTBMV-PM outperform all B&P results except for one instance. In 5 machines instances on the other hand, B&P algorithms perform better than FTBMV-PM, in 24 out of 25 instances. Moreover, the adaptive root node processing is able to provide better optimality gaps in 16 instances whereas the classical approach provides the best gap in 8 instances. We note that, with the increase in complexity of the run instance through number of resources and periods, the B&P algorithms tend to take more time to process.

However, it generates better solutions. The algorithm we propose, according to the numerical experiments, is capable of providing good quality solutions in more complex instances when MIP formulation FTBMV-PM fails to do so. However, FTBMV-PM appears to be more reliable option when there is a small number of parallel machines.

7.3. Multiple Level Network

7.3.1. Preliminary Tests on Family Based Model and B&P Extensions

In this Section, we present the results of preliminary tests on formulations extended to multiple level network instance. We will run the tests with FTBMV-MLN described in Section 6.1 and B&P algorithm described in Section 6.2. Similar to previous tests, we use a dataset corresponding to a realistic data set from the same float glass manufacturer. As we focus on multiple product level in multiple facilities and locations this Section, we now use the full dataset from the entire production network of the company. We run the experiments with 3 different data sets including 15 periods of planning horizon, and with color and coating in family structure. Finally, for the B&P algorithm, we use the same settings from parallel machine case. Namely, we generate unit extended patterns as P', employ DF for node selection and adaptive root node processing.

We present the results obtained with 1 hour limit in 7.13. We note that in these preliminary results, both FTBMV-MLN and B&P were unable to find an optimal solution within 1 hour time limit. In all instances, B&P outperforms FTBMV-MLN. In particular, B&P finds solutions with better incumbent objective values. However, B&P is unable to provide good lower bounds, hence the reported relative MIP gap is low. Moreover, FTBMV-MLN is also unable to obtain solutions better than 30% MIP gap on average. These preliminary results reveal that both FTBMV-MLN and B&P are unable to solve the problems to a reasonable quality within 1 hour. For the rest, we will focus on building approaches to produce good quality solutions, namely improved UB generation.

Table 7.13. Preliminary runs for Multiple Level Network.

	FTBMV-MLN			B&P		
Run	**Relative Gap**	**Objective**	**CPU Time (s)**	**Relative Gap**	**Objective**	**CPU Time (s)**
p:12-d:1	34.42%	665213804634	3600	77.12%	490165407954	3600
p:12-d:2	33.12%	656148881920	3600	78.06%	476991708944	3655
p:15-d:1	36.17%	730784938402	3600	80.18%	503774973280	3753

7.3.2. Data Set and Problem Instances

In the numerical experiments, similar to single and parallel machine instances, we use again a realistic dataset from the same major float glass manufacturer in Turkey. The family structure still consists of color and coating. Moreover, we noted in Section 5.4.4 the importance of testing the performance of suggested methods with datasets differing in terms of number of resources and periods. Since we already extend the problem to multiple product hierarchy and facilities, the number of resources are already overridden by this new network structure. Hence, we will focus in varying number of periods only.

We have two base datasets containing 67 locations, 20 production lines continuous and discrete combined, 373 products with 4748 product inventory combinations. The datasets are different from each other with respect to the number of periods; one has 12 periods whereas the other has 15 periods. Similar to parallel machine instances, for each of these base datasets, we have 5 different demand scenarios.

7.3.3. Preliminary Tests on Algorithms

In Section 7.3.1, preliminary tests revealed that FTBMV-MLN and CPM-EPMLN failed to provide good quality solutions within 1-hour limit, and therefore we decided to focus on matheuristics to obtain solution providing good quality UBs.

The new algorithms are based on problem size reduction through projecting demands to locations with production lines so that the problem becomes a multiple facility parallel machines campaign planning problem.

DPH and DDPH are in essence MIP models and we limit the solution time to 3600 seconds. As explained in Section 6.3.2, we solve the FTBMV-MLN with $\hat{\delta}$ variables fixed in order to obtain the objective function value in the original problem. Since this second model becomes an LP, the solution time is negligible. Since the DDPH-RNF is an iterative algorithm, we need to properly limit the solution time for each step such that the overall solution time does not exceed 3600 seconds. For that, we calculate the number of iterations based on (N^I, N^F) and split 3600 seconds equally to each iteration. Furthermore, we determine the parameters for (N^I, N^F) as $(1, 1)$, $(2, 2)$, $(3, 2)$, $(3, 3)$ and $(4, 2)$ to observe the sensitivity of short and longer horizons when fixing pattern decisions. In order to benchmark against the classical R&F, we also take runs applying R&F to FTBMV-MLN with the same set of parameters.

Table 7.14 shows the average outputs of preliminary test results. Similar to parallel machine preliminary tests, we randomly selected 3 instances. For each solution approach, we report the MIP optimality gap ("Gap" column) in addition to the UB obtained. Note that, it is not possible to report an optimality gap for the heuristics. Therefore, we use best LB obtained from running FTBMV-MLN for the same instances. *Algorithm* column in the table shows the solution algorithm with a smart representation. For DDPH instances, a stands for the number of alternatives made available in the model, e.g. DDPH-a:3 means there are 3 demand projecrtion alternatives available. Finally, for RNF and DDPH-RNF, i stands for parameter N^I and f for parameter N^F.

Results show that for demand projection heuristics without R&F, extra alternatives help with the UB. DDPH consistently generate better UB values then DPH. Moreover, between having 2 or 3 alternatives does not result in much difference in the objective value and the calculated gap with 2-alternatives model being better.

Table 7.14. Preliminary test results on algorithms.

Method	**Calculated Gap**	**CPU Time**	**Objective**
DPH	3.45%	3600	483086117111.16
DDPH-a:2	2.24%	3600	477127171537.39
DDPH-a:3	2.24%	3600	477127171537.39
RNF-i:1-f:1	100.00%	3600	#N/A
RNF-i:2-f:2	100.00%	3600	#N/A
RNF-i:3-f:2	3.91%	2100	485397993834.10
RNF-i:3-f:3	11.03%	1740	524230623554.69
RNF-i:4-f:2	2.52%	2700	478499108441.36
DDPH-RNF-i:1-f:1	9.28%	1264	522379559102.66
DDPH-RNF-i:2-f:2	2.21%	1372	476959291273.18
DDPH-RNF-i:3-f:2	2.17%	2244	476760695505.81
DDPH-RNF-i:3-f:3	2.58%	1699	478783994395.82
DDPH-RNF-i:4-f:2	1.92%	2665	475530622687.18

Additionally, 3-alternatives model has more decision variables. Classical R&F algorithms fail to generate any solution within 3600 seconds with parameters $(1, 1)$ and $(2, 2)$ for (N^I, N^F). This is mainly due to time spent at root node preprocessing in iterations, which leads to the termination of the algorithm without obtaining any integer solution within time limit. Moreover, with other parameters, R&F still is outperformed by DDPH-RNF solutions. We observe that, for DDPH-RNF generates the best quality UB values in shorter time than the time limit. Regarding other parameter sets, none of them significanlty outperforms the others. Although $(1, 1)$ seems to be low in quality, when we drill down in detail, one specific instance with 24% calculated gap is the root cause for the lower gap than other parameters. Following these preliminary tests, for the remainder of our numerical experiments, we will continue with following algorithms: DDPH-a:2, DDPH-RNF with parameters all the parameters tested.

7.3.4. Multiple Level Network Numerical Results

Similar to our previous tests in parallel machine results and multiple level network preliminary tests, we limit the overall solution time to 1-hour and employ the limiting strategy for iterative R&F combined algorithms. We run the experiments with 5 different datasets and for 12 and 15 planning periods.

Tables 7.15 and 7.16 show the outputs of numerical results for 12 and 15 period datasets respectively. We report the MIP optimality gap calculated with the best LB value obtained from running FTBMV-MLN, objective value and CPU time spent for each algorithm. Note that, similar to previous results sections, *Run* column is a smart representation. It indicates the number of available projection alternatives with a, and the number of integer and fixed periods with i and f respectively. Finally, d stands for the dataset id for each run. We will compare the results similar to the approach described in Section 7.1.2, where an algorithm outperforms the others if it obtains a solution with lower UB.

In 12 period instances, DDPH-RNF with $(1, 1)$ outperforms others in 3 out of 5 instances. DDPH with 2 alternatives and DDPH-RNF with $(1, 1)$ both are best performants in 1 instance. Regarding the CPU times, DDPH-RNF with $(2, 2)$ is able to provide an UB in 754 seconds on average, with significantly low run time with 673 seconds for dataset 3. The average calculated optimality gap is 1,48% being close to average of 1,42% of DDPH-RNF with $(2, 2)$. DDPH-RNF with $(3, 3)$ is the algorithm that provided the quickest UB across all datasets with a total run time of 638 seconds. Another observation regarding the run times is related to the synchronization of N^I and N^F parameters. The instances $(3, 2)$ and $(4, 2)$ seem to consistently have longer running times.

In 15 period instances, an immediate observation is the longer running times compared to 12 period instances. Regarding the quality of the UB, DDPH-RNF with $(1, 1)$ outperforms all other methods in 4 out of 5 instances. For the remaining instance, i.e. dataset 5, DDPH-RNF with $(4, 2)$ is able to provide an UB with 2% of calculated gap. The DDPH-RNF with $(1, 1)$ was able to provide only a 24% calculated optimality gap for this specific instance. As explained in previous Section, the longer node preprocessing time spent by the solver in iterations seemed to be the cause when we analyze the optimization log messages. We conclude that DDPH-RNF being able to provide good quality UB values, hence integral solutions, with low run times is promising for solving MLN campaign planning problem.

Table 7.15. Results for instances with 12 periods.

Run	Objective	Calculated Gap	CPU Time(s)
DDPH-a:2-d:1	440967615996.18	1.02%	3600
DDPH-a:2-d:2	445818037614.20	1.57%	3600
DDPH-a:2-d:3	442722918863.90	1.86%	3600
DDPH-a:2-d:4	444344475244.95	1.55%	3600
DDPH-a:2-d:5	443524737467.42	1.55%	3600
DDPH-RNF-i1-f:1-d:1	442528480402.77	1.37%	980
DDPH-RNF-i1-f:1-d:2	445041414435.98	1.39%	914
DDPH-RNF-i1-f:1-d:3	441085548415.43	1.50%	901
DDPH-RNF-i1-f:1-d:4	443717427310.98	1.41%	935
DDPH-RNF-i1-f:1-d:5	442853869579.62	1.40%	960
DDPH-RNF-i2-f:2-d:1	442905568597.18	1.45%	705
DDPH-RNF-i2-f:2-d:2	445432413294.81	1.48%	968
DDPH-RNF-i2-f:2-d:3	441076203317.50	1.50%	673
DDPH-RNF-i2-f:2-d:4	444082011329.05	1.49%	713
DDPH-RNF-i2-f:2-d:5	443244675850.15	1.49%	710
DDPH-RNF-i3-f:2-d:1	444714455629.32	1.85%	1261
DDPH-RNF-i3-f:2-d:2	447232849136.53	1.88%	1087
DDPH-RNF-i3-f:2-d:3	442882410146.80	1.90%	1330
DDPH-RNF-i3-f:2-d:4	445878042163.24	1.89%	1371
DDPH-RNF-i3-f:2-d:5	445866914921.08	2.07%	1344
DDPH-RNF-i3-f:3-d:1	445398547416.54	2.00%	801
DDPH-RNF-i3-f:3-d:2	447928050439.39	2.03%	648
DDPH-RNF-i3-f:3-d:3	443585814183.06	2.06%	703
DDPH-RNF-i3-f:3-d:4	446086652562.20	1.94%	787
DDPH-RNF-i3-f:3-d:5	445745017969.11	2.04%	1400
DDPH-RNF-i4-f:2-d:1	444714334614.87	1.85%	2948
DDPH-RNF-i4-f:2-d:2	447331390547.57	1.90%	2781
DDPH-RNF-i4-f:2-d:3	442882405119.92	1.90%	2794
DDPH-RNF-i4-f:2-d:4	445878037263.13	1.89%	2826
DDPH-RNF-i4-f:2-d:5	444947266391.36	1.86%	2562

Table 7.16. Results for instances with 15 periods.

Run	Objective	Calculated Gap	CPU Time(s)
DDPH-a:2-d:1	475448308190.12	1.98%	3600
DDPH-a:2-d:2	477135685517.33	2.14%	3600
DDPH-a:2-d:3	474395827910.45	1.78%	3600
DDPH-a:2-d:4	476685957865.65	1.97%	3600
DDPH-a:2-d:5	479001219346.38	2.44%	3600
DDPH-RNF-i1-f:1-d:1	473652870157.28	1.61%	1814
DDPH-RNF-i1-f:1-d:2	475823201336.41	1.87%	1814
DDPH-RNF-i1-f:1-d:3	473636165725.66	1.62%	1849
DDPH-RNF-i1-f:1-d:4	475443844777.58	1.71%	2357
DDPH-RNF-i1-f:1-d:5	617662605814.28	24.34%	1213
DDPH-RNF-i1-f:1-d:1	476548862482.78	2.21%	1311
DDPH-RNF-i2-f:2-d:2	477491277867.18	2.21%	1111
DDPH-RNF-i2-f:2-d:3	476485936838.88	2.21%	1192
DDPH-RNF-i2-f:2-d:4	477868777496.69	2.21%	1349
DDPH-RNF-i2-f:2-d:5	477895209268.78	2.21%	1283
DDPH-RNF-i3-f:2-d:1	478248865229.77	2.55%	2044
DDPH-RNF-i3-f:2-d:2	478241367852.03	2.37%	2441
DDPH-RNF-i3-f:2-d:3	476346538860.48	2.18%	2418
DDPH-RNF-i3-f:2-d:4	479468033121.02	2.54%	2071
DDPH-RNF-i3-f:2-d:5	479492303946.51	2.54%	1816
DDPH-RNF-i3-f:3-d:1	478229745439.84	2.55%	1746
DDPH-RNF-i3-f:3-d:2	478908207358.55	2.50%	1821
DDPH-RNF-i3-f:3-d:3	477221319263.33	2.36%	1829
DDPH-RNF-i3-f:3-d:4	477600752553.65	2.15%	2016
DDPH-RNF-i3-f:3-d:5	479718413430.60	2.58%	1678
DDPH-RNF-i4-f:2-d:1	480039967785.19	2.92%	3116
DDPH-RNF-i4-f:2-d:2	481830905083.43	3.09%	3277
DDPH-RNF-i4-f:2-d:3	475453892546.52	2.00%	3006
DDPH-RNF-i4-f:2-d:4	477595713775.92	2.15%	2672
DDPH-RNF-i4-f:2-d:5	476847730690.80	2.00%	2979

8. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this Chapter, we focus on future research directions regarding the campaign planning problem under sequence-dependent family setups and co-production, and the instances we worked on, namely single machine, parallel machines and multiple level network. The solution methods we propose can also be adapted and applied to other process industries such as food and beverage, oil refineries, various chemical processes of pharmaceuticals etc. Extension to oil industries can particularly be beneficial since it also has co-production phenomenon. We share some general notes on the proposed solutions as well as summarizing and pointing some potential improvement areas.

The mathematical formulations and methods are suitable for practical use as decision support tool in practice. We note that, even though the last instance that is multiple product hierarchy and network structure is the environment for most float glass companies, there exist also manufacturers with single or parallel machines in a single facility. Hence, the practical implication of each method separately is pertinent. Moreover, the methods are able to cope with uncertainty up to a certain level. For demand uncertainty, the planners can use methods for different demand scenarios and combine the output campaign plans. Co-production is another source of uncertainty in terms of production decisions and all the methods aim to provide flexibility for planners to comply the realizations against the plans.

In single machine instance, we proposed two MIP formulations and their variants. Numerical results revealed that, Family Transition Based Models outperform Pattern Transition Based Models. We investigated the effects of family structure and period dimensions to the formulation characteristics and solution performance. Pattern Based Models become much larger in terms of number of variables and constraints with the increase in these dimensions. Moreover, Family Based Models provide higher LP relaxation objectives in all test instances.

Tests reveal that Family Based Models are able to provide optimal solution for all instances. Run times on all numerical studies were ideal for a single machine instance in practice. However, there seems to be still room for improvement to further reduce. According to the optimization logs, models actually find an optimal solution earlier than reported run time, which means there is difficulty in proving the optimality. Introducing cuts exploiting costs would help improving the lower bound.

We provide additional analysis to analyze the sensitivity of the costs. Increasing setup costs increases average campaign duration in most cases parallel to the expectation. We also provide some further insights that shows in combination with inventory holding costs of different products, this trend can also change. In accordance with longer campaigns, amount of total inventory carried also increases. Considering backlog cost increase, average inventory carried also increases, which shows that the model tries to avoid backlog as much as possible. Finally, inventory holding cost shows the least impact on the campaign decisions since it is outdone by setup and demand satisfaction related costs. Such analysis provide business insights to the behavior of the models.

In parallel machine instance, we continue with the most complex instance of the single machine case. We observe that, Pattern Based Models' performance deteriorates even more, and as a result we do not further investigate them as a viable solution option. As per Family Base Models, they are significantly better. However, they lose efficiency when the problem becomes more and more complex with increased number of parallel machines. The novel B&P algorithm is capable of providing better quality solutions for those cases. However, in some instances both the models struggle with improving MIP gap to under 5%.

We note that we limit the number of threads to one for both the methods to have a fair comparison. However, this is a limitation and it would be a solid next step to convert the B&P algorithm to parallel node processing mode. Moreover, we also observe a similar behavior to single machine case when we analyze the logs.

The lower bound improvement seems as a promising next step. Family Based Models could benefit from symmetry breaking constraints in the existence of identical machines. Another promising idea is to improve node selection criteria for B&P algorithm. Commercial solvers employ sophisticated methods such as keeping track of pseudo-costs for each node and using them as an indicator for likelihood of providing good solutions. A similar approach could also be beneficial for B&P algorithm. Last but not least, root node processing with C&G takes long time in some instances. Providing better initial column set could help with this issue.

Finally, in network model we further extend the problem to multiple product hierarchy and multiple facilities, to a network in other words. We extend the FTBMV and B&P algorithm. However, preliminary tests show that these exact optimization approaches fail to provide good quality solutions within time limits. Hence, we focus on improving UB generation, namely integral solutions. We exploit the business insights from the problem definition and reduce the problem again to float glass campaign planning problem on multiple facilities by projecting demands to facilities. Numerical experiments reveal that the 3 variants of the demand projection heuristics are able to provide good quality UBs, with an indication from calculated MIP gaps using best known LBs. Considering satisfactory run times obtained with demand projection heuristics, a further improvement can be using these solutions as MIP warm start and let the original problem to run until 1-hour limit. This can help with obtaining the real MIP gaps, further improve the incumbent value or even prove the solution optimal. Additionally, we can formulate another iterative heuristic algorithm exploiting the concept of extended patterns and demand projection. Given an extended pattern for each resource, we can determine optimal allocation of resource capacities to demands with an LP based on families. The second part will then focus on determining new extended patterns given the allocation of demands to resources, based on a local search or another pricing problem similar to the one explained in Section 5.4.4 that will modify the durations of families within the extended pattern.

REFERENCES

1. Glass, G. P. F., *Float Glass - Gold Plus Glass*, 2019, `https://goldplusgroup.com/float-glass/`, accessed in September 2020.

2. Taşkın, Z. C. and A. T. Ünal, "Tactical Level Planning in Float Glass Manufacturing with Co-Production, Random Yields and Substitutable Products", *European Journal of Operational Research*, Vol. 199, No. 1, pp. 252–261, 2009.

3. Almada-Lobo, B., J. F. Oliveira and M. A. Carravilla, "Production planning and scheduling in the glass container industry: A VNS approach", *International Journal of Production Economics*, Vol. 114, pp. 363–375, 2008.

4. Toledo, C. F. M., M. da Silva Arantes, M. Y. B. Hossomi and B. Almada-Lobo, "Mathematical programming-based approaches for multi-facility glass container production planning", *Computers and Operations Research*, Vol. 74, pp. 92–107, 2016.

5. Haase, K. and A. Kimms, "Lot sizing and scheduling with sequence-dependent setup costs and times and effcient rescheduling opportunities", *International Journal of Production Economics*, Vol. 66, pp. 159–169, 2000.

6. Fiorotto, D. J., R. Jans and S. A. de Araujo, "An analysis of formulations for the capacitated lot sizing problem with setup crossover", *Computers and Industrial Engineering*, Vol. 106, pp. 338–350, 2017.

7. Wittrock, R., "Scheduling Parallel Machines with Major and Minor Setup Times", *The International Journal of Flexible Manufacturing Systems*, Vol. 2, pp. 329–341, 1990.

8. Stefansdottir, B., M. Grunow and R. Akkerman, "Classifying and modeling setups and cleanings in lot sizing and scheduling", *European Journal of Operational*

Research, Vol. 261, pp. 849–865, 2017.

9. Ağralı, S., "A dynamic uncapacitated lot-sizing problem with co-production", *Optimization Letters*, Vol. 6, pp. 1051–1061, 2012.

10. Florian, M., J. K. Lenstra and A. H. G. R. Kan, "Deterministic Production Planning: Algorithms and Complexity", *Management Science*, Vol. 26, No. 7, pp. 669–679, 1980.

11. Fleischmann, B. and H. Meyr, "The general lotsizing and scheduling problem", *OR Spectrum*, Vol. 19, pp. 11–21, 1997.

12. Copil, K., M. Worbelauer, H. Meyr and H. Tempelmeier, "Simultaneous lotsizing and scheduling problems: a classification and review of models", *OR Spectrum*, Vol. 39, pp. 1–64, 2017.

13. Allahverdi, A., C. T. D. Ng, T. C. E. Cheng and M. Kovalyov, "A survey of scheduling problems with setup times or costs", *European Journal of Operational Research*, Vol. 18, pp. 985–132, 2008.

14. Allahverdi, A., "The third comprehensive survey on scheduling problems with setup times or costs", *European Journal of Operational Research*, Vol. 246, pp. 345–378, 2015.

15. Herr, O. and A. Goel, "Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints", *European Journal of Operational Research*, Vol. 248, pp. 123–135, 2016.

16. Jin, F., S. Song and C. Wu, "A simulated annealing algorithm for single machine scheduling problems with family setups", *Computers and Operations Research*, Vol. 36, pp. 2133–2138, 2009.

17. Uzsoy, R. and J. D. Vel¡asquez, "Heuristics for minimizing maximum lateness on

a single machine with family-dependent set-up times", *Computers and Operations Research*, Vol. 35, pp. 2018–2033, 2008.

18. Guo, Q. and L. Tang, "An improved scatter search algorithm for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times", *Applied Soft Computing*, Vol. 29, pp. 184–195, 2015.

19. Gören, H. G. and S. Tunalı, "Solving the capacitated lot sizing problem with setup carryover using a new sequential hybrid approach", *Applied Intelligence*, Vol. 42, pp. 805–816, 2015.

20. Araujo, S. A. and A. Clark, "A priori reformulations for joint rolling-horizon scheduling of materials processing and lot-sizing problem", *Computers and Industrial Engineering*, Vol. 65, pp. 577–585, 2013.

21. Toso, E. A., R. Morabito and A. Clark, "Lot sizing and sequencing optimisation at an animal-feed plant", *Computers and Industrial Engineering*, Vol. 57, pp. 813–821, 2009.

22. Clark, A., R. Morabito and E. A. Toso, "Production setup-sequencing and lot-sizing at an animal nutrition plant through ATSP subtour elimination and patching", *Journal of Scheduling*, Vol. 13, pp. 111–121, 2010.

23. Günther, H. O., "The block planning approach for continuous time-based dynamic lot sizing and scheduling", *Business Research*, Vol. 7, pp. 51–76, 2014.

24. Figueira, G., M. O. Santos and B. Almada-Lobo, "A hybrid VNS approach for the short-term production planning and scheduling: A case study in the pulp and paper industry", *Computers and Operations Research*, Vol. 40, pp. 1804–1818, 2013.

25. Furlan, M., B. Almada-Lobo, M. Santos and R. Morabito, "Unequal individual genetic algorithm with intelligent diversification for the lot-scheduling problem in integrated mills using multiple-paper machines", *Computers and Operations*

Research, Vol. 59, pp. 33–50, 2015.

26. Figueira, G., P. Amorim, L. Guimarães, M. Amorim-Lopes, F. Neves-Moreira and B. Almada-Lobo, "A decision support system for the operational production planning and scheduling of an integrated pulp and paper mill", *Computers and Chemical Engineering*, Vol. 77, pp. 85–104, 2015.

27. Toledo, C. F. M., M. da Silva Arantes, R. R. R. de Oliveira and B. Almada-Lobo, "Glass container production scheduling through hybrid multi-population based evolutionary algorithm", *Applied Soft Computing*, Vol. 13, pp. 1352–1364, 2013.

28. Almada-Lobo, B., D. Klabjan, Maria, A. Carravilla and J. F. Oliveira, "Multiple machine continuous setup lotsizing with sequence-dependent setups", *Computational Optimization and Applications*, Vol. 47, pp. 529–552, 2010.

29. Farahani, P., M. Grunow and H. O. Günther, "Integrated production and distribution planning for perishable food products", *Flexible Services and Manufacturing*, Vol. 24, pp. 28–51, 2012.

30. Joly, M., L. F. L. Moro and J. M. Pinto, "Planning and scheduling for petroleum refineries using mathematical programming", *Brazilian Journal of Chemical Engineering*, Vol. 19, pp. 207–228, 2002.

31. Schaller, J., "Minimizing total tardiness for scheduling identical parallel machines with family setups", *Computers and Industrial Engineering*, Vol. 72, pp. 274–281, 2014.

32. Zhao, Z. and X. Li, "Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming", *Operations Research for Health Care*, Vol. 3, pp. 160–167, 2014.

33. Kramer, A., M. Iori and P. Lacomme, "Mathematical formulations for scheduling

jobs on identical parallel machines with family setup times and total weighted completion time minimization", *European Journal of Operational Research*, 2019, https://doi.org/10.1016/j.ejor.2019.07.006.

34. Hinder, O. and A. J. Mason, "A novel integer programing formulation for scheduling with family setup times on a single machine to minimize maximum lateness", *European Journal of Operational Research*, Vol. 262, pp. 411–423, 2017.

35. James, R. J. and B. Almada-Lobo, "Single and parallel machine capacitated lot-sizing and scheduling: New iterative MIP-based neighborhood search heuristics", *Computers and Operations Research*, Vol. 38, pp. 1816–1825, 2011.

36. Camargo, V., F. Toledo and B. Almada-Lobo, "Three time-based scale formulations for the two-stage lot sizing and scheduling in process industries", *Journal of the Operational Research Society*, Vol. 63, pp. 1613–1630, 2012.

37. Xue, G., F. Offodile, H. Zhou and M. Troutt, "Integrated production planning with sequence-dependent family setup times", *International Journal of Production Economics*, Vol. 131, pp. 674–681, 2011.

38. Camargo, V., F. Toledo and B. Almada-Lobo, "HOPS – Hamming-Oriented Partition Search for production planning in the spinning industry", *European Journal of Operational Research*, Vol. 234, pp. 266–277, 2014.

39. Rafiei, R., M. Nourelfath, J. Gaudreault, L. A. D. Santa-Eulalia and M. Bouchard, "Dynamic safety stock in co-production demand-driven wood remanufacturing mills: A case study", *International Journal of Production Economics*, Vol. 165, pp. 90–99, 2015.

40. Seeanner, F., B. Almada-Lobo and H. Meyr, "Combining the principles of variable neighborhood decomposition search and the fix&optimize heuristic to solve multi-level lot-sizing and scheduling problems", *Computers and Operations Research*,

Vol. 40, pp. 303–317, 2013.

41. Gaudreault, J., J.-M. Frayret, A. Rousseau and S. D'Amours, "Combined planning and scheduling in a divergent production system with co-production: A case study in the lumber industry", *Computers and Operations Research*, Vol. 38, pp. 1238–1250, 2011.

42. Ghirardi, M. and A. Ameiro, "Matheuristics for the lot sizing problem with back-ordering, setup carryovers, and non-identical machines", *Computers and Industrial Engineering*, Vol. 127, pp. 822–831, 2019.

43. Bektur, G. and T. Saraç, "A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server", *Computers and Operations Research*, Vol. 103, pp. 46–63, 2019.

44. Vieira, M., T. Pinto-Varela, S. Moniz, A. P. Barbosa-Póvoa and L. G. Papageorgiou, "Optimal planning and campaign scheduling of biopharmaceutical processes using a continuous-time formulation", *Computers and Chemical Engineering*, Vol. 91, pp. 422–444, 2016.

45. Susarla, N. and I. Karimi, "Integrated campaign planning and resource allocation in batch plants", *Computers and Chemical Engineering*, Vol. 35, pp. 2990–3001, 2011.

46. Dhaenens-Flipo, C. and G. Finke, "An integrated model for an industrial production distribution problem", *IIE Transactions*, Vol. 33, pp. 705–715, 2001.

47. Wei, W., L. Guimarães, P. Amorim and B. Almada-Lobo, "Tactical production and distribution planning with dependency issues on the production process", *Omega*, Vol. 67, pp. 99–114, 2017.

48. Guimarães, L., D. Klabjan and B. Almada-Lobo, "Modeling Lot Sizing and

Scheduling Problems with Sequence Dependent Setups", *European Journal of Opreations Research*, Vol. 239, pp. 644–662, 2014.

49. Miegeville, N., *Supply Chain Optimization in the Process Industry. Methods and and case-study of the glass industry.*, Ph.D. Thesis, Ecole Centrale, Paris, 2005.

50. Lime, R., I. Grossmann and Jiao, "Long-term scheduling of a single-unit multi-product continuous process to manufacture high performance glass", *Computers and Checmial Engineering*, Vol. 35, pp. 554–574, 2011.

51. Öner, S. and T. Bilgiç, "Economic lot scheduling with uncontrolled co-production", *European Journal of Operational Research*, Vol. 188, pp. 793–810, 2008.

52. Kopanos, G., L. Puigjaner and C. Maravelias, "Production Planning and Scheduling of Parallel Continuous Processes with Product Families", *Industrial Engineering and Chemistry Research*, Vol. 50, pp. 1369–1378, 2011.

53. Brierley, J., C. Cowton and C. Drury, "A comparison of products costing practices in discrete-part and assembly manufacturing and continuous production process manufacturing", *International Journal of Production Economics*, Vol. 100, pp. 314–321, 2006.

54. Ahuja, R. K., T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.