PLANNING AND SCHEDULING IN PHARMACEUTICAL SUPPLY CHAINS

NARESH SUSARLA

(BE in Chemical Engg., National Institute of Technology, Durgapur, India)

A THESIS SUBMITTED

FOR THE DEGREE OF PHD OF ENGINEERING

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

To my father, mother, ravi, and chelli

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

S. Naresh

Naresh Susarla

15 August 2012

ACKNOWLEDGEMENTS

On the threshold of my career, I express my sincere gratitude and love to my parents (Mr Sivaprasad and Mrs Balagowri) for their unconditional faith and support that led me until now. As no battle can be won fighting alone, in my conquest of a PhD degree, I was continuously supported by some truly remarkable and inspiring souls. It is with immense pleasure and respect I take this opportunity to express my gratitude to all those who have helped me in shaping my research career and making my stay in Singapore a truly memorable one.

To begin a career in research with absolutely no inkling of what it takes, the carefully carved guidance of Prof. I. A. Karimi was a little more than what I would have ever expected in my rarest of dreams. With enormous respect, I express my sincere gratitude to him for his continuous encouragement and inspiration. Words fail to express my feelings in thanking him for his immense support, timely inputs, and invaluable suggestions that played crucial role in shaping this thesis. There is no way in which I can thank Prof Karimi for the rich research environment he provided me in the past five years that help me develop my personality in myriad ways. It was indeed a privilege and a personal delight to have worked with him.

I extend my sincere thanks to Prof. Raj and Prof. Laksh for their kind acceptance to be on the panel of examiners during the qualifying exam. Their insights and invaluable suggestions were instrumental in deciding the course of this research. I truly appreciate their kind encouragement and guidance at various occasions during the course of my PhD. I also thank the final reviewers for spending time on evaluating this thesis. I wish to admire and thank all the unknown reviewers of our publications, who gave constructive feedbacks on all our manuscripts and helped us to bring the best out of this research.

I am extremely thankful to Mr. Allan Chin, Mr. Lim Eng Seng, and Ms. Caroline from GSK, Singapore for providing me the opportunity to work with them and helping me in identifying the real-life problems and industrial challenges those we later tried to address in this thesis. It was indeed a remarkable experience to directly apply my research to a real life scenario and assess its effectiveness. I also express my gratitude to all the professors, whose valuable lectures/seminars have been the stepping stones in building up ideas.

There are no words to thank my beloved friends Abhik, Suresh, Shilpi, Suvankar, Vinayak, Abhinav, Kiran, Arun, Rakesh, Reeto, Kamnashish, Nikhil, Saurav, and more, who stood by me when the times were tough. This journey was not possible without their kind and loving support. Special thanks to my labmates (Sadegh, Li Jie, Sangeeta, Faruque, Vasanth, Kefeng, Anoop, Rajnish, Karthik, Mona, Hanifah, Razib, Suresh) and friends at NUS (Kaushik, Vaibhav, Sumit, Shivom, Krishna, KMG, Thanneer, Srinath, Manoj, Saran, Satyanarayana, Sujit, and more), who made my stay in Singapore and at NUS all the more beautiful and enjoyable.

And most importantly, I thank my brother Ravi, sister Sharada, my cousins, and other family members for always being my source of inspiration. Their love, continuous support, and motivation have been really helpful.

TABLE OF CONTENTS

SUMMARY

The global pharmaceutical industry is grappling with tremendous turmoil in the marketplace and dramatically changing competitive landscape. Fierce market competition, peaking patent cliffs, mounting R&D costs, shrinking product pipelines and stringent regulatory protocols have brought a complete paradigm shift in the way pharmaceutical enterprises operate. To meet these challenges and remain competitive in the market, companies seek cutting edge technologies for better management of operations, handling resources, and reducing costs. This PhD work identifies and addresses a number of critical challenges in supply chain operations and managerial decision making for pharmaceutical companies.

First, we studied and analysed some of the recent multi-grid batch scheduling models. Here, we identified the limitations of the existing multi-grid approaches and suggested ways to address such limitations. Motivated with this study, we developed two novel multi-grid continuous-time formulations for scheduling multipurpose batch plants. Their major contributions are a fool-proof and novel use of unit-slots in managing shared resources such as materials and the flexibility to allow non-simultaneous transfers of materials into a batch. However, the real plant operations usually involve several other resources (human, utilities, etc.) and additional characteristics such as sequence-dependent transition times, non-zero material transfer times, etc. Thus, we extended our approach of unit-slots to consider a more realistic scheduling problem. Also, we present a new and a more comprehensive single-grid model, which considers many of the aforementioned characteristics and compare against a number of other single-grid and multi-grid models.

We then extended our aforementioned study of resource constrained scheduling to consider a bigger and integrated problem of production planning and resource allocation (specifically) in pharmaceutical plants. Here, we developed a framework to capture the key aspects of the industrial planning activity such as interactions among the planner and other stakeholders and the effect of resource allocation on process performance. We presented a novel treatment for key aspects of an industrial planning such as maintenance, NPIs, resource allocations, safety stock, delivery delays, etc., and gives the exact number of batches and schedule for each campaign. Also, we demonstrated the usefulness of our model using two realistic examples.

Next, we extended our study of production planning of a pharmaceutical plant to the operational planning of the entire production supply chain. Here, we considered entire functions of an enterprise from procurement of raw materials to distribution of final products in a seamless fashion with a granularity of individual processing tasks and campaigns on production lines. The focus has primarily been on the development of a simple model that is easy to implement, quick to solve, and does not compromise on the realism or features of the problem. Our model incorporates several practical features of industrial planning such as effects of international tax differentials, inventory holding costs, material shelf-lives, waste treatment / disposal, and other real-life factors on the after-tax profit of a company.

Finally, we presented a tool for the integrated production planning and resource allocation in pharmaceutical plants. Here, we highlighted the limitations of the existing technologies and established the necessary features for such a tool. We have further discussed and incorporated some of the real-life challenges and industrial practices important for the industrial planners in our tool.

NOMENCLATURE

Chapter 3

Notation

Indices

i	task
j	unit
n	event
r	resource
s	state
S	storage

Sets

I	tasks
I_p	processing tasks
I_r	tasks related to resource r
$suit_{ij}$	tasks that can be performed in unit j
J	units (both processing and storage)
N	event points within the time horizon
R	resources
S	states

Parameters

ρ_{si}^{p}, ρ_{si}^{c}, ρ_{ij}	Proportion of state s produced, consumed from tasks i respectively,
ρ_{si}^{p}	≥ 0, $\rho_{si}^{c} \leq 0$

ρ_{ri}^{p}, ρ_{ri}^{c}	Proportion of material resource r produced, consumed from tasks i respectively, $\rho_{ri}^{p} \geq 0$, $\rho_{ri}^{c} \leq 0$
μ_{ri}^{p}, μ_{ri}^{c}	Proportion of equipment resource r produced, consumed from tasks i respectively, $\mu_{ri}^{p} \geq 0$, $\mu_{ri}^{c} \leq 0$

Variables

$b(i, n)$	amount of material undertaking task i at event n, ton
$b(i, j, n)$	amount of material undertaking task i in unit j at event n, ton
$E_0(r)$	initial amount of resource r available or required from external sources
$E(r, n)$	excess amount of resource r available at event n
$ST_0(s, n)$	Amount of state $s \in S^R$ that is required from external resources at event n, ton
$ST(s, n)$	excess amount of state s that needs to be stored at event n, ton
$w(i, j, n)$	binary variable for assignment of task i in unit j at the beginning of event n
$w(i, n)$	binary variable for assignment of task i at the beginning of event n

Chapter 4

Notation

Indices

s	material state
i	task
j	unit
k	slot

Superscripts

L lower limit

U upper limit

Sets

$\mathbf{I}_j$ Tasks that unit j can perform

Parameters

σ_{sij} Mass Ratio for material s in task i on unit j

H Scheduling Horizon

υ_s Unit price of s

α_{ij} Parameter for determining the processing time of task i on unit j

β_{ij} Parameter for determining the processing time of task i on unit j

d_s Minimum demand for material s

Variables

T_{jk} Time at which slot k on unit j ends

T_{sk} Time at which slot k on storage s ends

BI_{ijk} Amount of task i entering unit j at the start of slot k

ΔBI_{ijk} Amount above the minimum batch size in BI_{ijk}

b_{ijk} Amount of task i that resides in unit j at the end of slot k

BO_{ijk} Batch material output by task i at its completion within slot k

t_{jk} Time remaining at T_{jk} to complete the task in progress in slot k on unit j

I_{sk} Inventory level of s at T_{jk}

δ_{jk} Time period for which unit j idles in the beginning of slot k

θ_{jk} Time period for which unit j idles towards the end of slot k

Binary

ys_{ijk}	1, if unit j begins a task at the beginning of slot k

0-1 Continuous

z_{jk}	1, if unit j ends an ongoing task within slot k
ye_{ijk}	1, if unit j ends a task i within slot k
y_{ijk}	1, if unit j continues task i at time T_{jk}

Chapter 5

Notation

Indices

s	material state
i	task
j	unit
k	slot
r	resource
o	operation

Superscripts

L	lower limit
U	upper limit

Sets

$\mathbf{I}_j$	Tasks that unit j can perform
$\mathbf{I}_r$	Tasks that utilize resource r

Parameters

σ_{sij}	Mass Ratio for material s in task i on unit j
H	Scheduling Horizon
υ_s	Unit price of s

α_{ij}	Parameter for determining the processing time of task *i* on unit *j*
β_{ij}	Parameter for determining the processing time of task *i* on unit *j*
μ_{ri}	Parameter for determining the consumption of resource *r* by task *i*
ν_{ri}	Parameter for determining the consumption of resource *r* by task *i*
γ_{sj}	Parameter for determining the transfer time for material *s* on unit *j*
$q0_s$	Initial available inventory for material *s*
D_s	Minimum demand for material *s*

Variables

T_k	Time at which process slot *k* ends
t_{jk}	Time at which slot *k* on unit *j* ends
t_{sk}	Time at which slot *k* on storage *s* ends
t_{rk}	Time at which slot *k* on resource *r* ends
b_{ijk}	Amount of task *i* entering unit *j* at the start of slot *k*
Δb_{ijk}	Amount above the minimum batch size in BI_{ijk}
br_{ijk}	Amount of task *i* that resides in unit *j* at the end of slot *k*
be_{ijk}	Batch material output by task *i* at its completion within slot *k*
tp_{jk}	Time remaining at t_{jk} to complete the task in progress in slot *k* on unit *j*
q_{sk}	Inventory level of *s*
π_{ij}	Cleaning time during slot *k* on unit *j*
δ_{jk}	Time period for which unit *j* idles in the beginning of slot *k*
θ_{jk}	Time period for which unit *j* idles towards the end of slot *k*
sl_k	Length of slot *k*
as_{sijk}	Amount of material *s* transferred from *s* to *j* or vice-versa

Binary

ys_{ijk}	1, if unit j begins allocation of a task at the beginning of slot k
ys_{iojk}	1, if unit j begins allocation of operation o of task in slot k
yts_{sijk}	1, if transfer of s for task i at the beginning of slot k

0-1 Continuous

x_{ijk}	1, if i is the latest/current task in slot k of unit j
ye_{ijk}	1, if unit j ends a task i within slot k
ye_{iojk}	1, if unit j ends an operation of task i within slot k
yte_{sijk}	1, if transfer of s for a task i ends in slot k of unit j
yr_{iojk}	1, if unit j continues operation o of task i at time T_k
ytr_{sijk}	1, if transfer of s for i continues from k to k + 1 in unit j

Chapter 6

Notation

Indices

s	material state
i	task
j	unit
r	resource
k	slot
t	time intervals

Superscripts

L	lower limit
U	upper limit

Sets

$\mathbf{I}_j$	Tasks that unit j can perform

$\mathbf{I}_r$	Tasks that consume resource r
$\mathbf{K}_j$	Number of slots per interval for unit j
$\mathbf{K}_r$	Number of slots per interval for resource r

Parameters

σ_{si}	Mass Ratio for material s in task i
DD_t	Product delivery date
h_t	Length of each interval
υ_s	Unit price of s
$\delta_{ii'}$	Parameter for delay between the starts of i and i'
$\tau_{ii'}$	Sequence-dependent changeover time between i and i'
pt_{ij}	Processing time of task i on unit j
ct_{ij}	Cycle time of task i on unit j
b_{ij}	batch size of task i on unit j
D_{st}	Minimum demand for material s
SS_{st}	Safety stock for material s in interval t
γ_{ijr}	Usage of resource r by task i on unit j

Continuous Variables

T_{jkt}	Time at which slot k on unit j ends in t
T_{rkt}	Time at which slot k on resource r ends in t
Q_{st}	Inventory level of s at the end of t
DO_{st}	Unsatisfied demand of material s at the end of t
S_{st}	Amount of s supplied to customers at the end of t
OS_{st}	Amount of s outsourced from 3rd party in t
RL_{ijkt}	Run length of task i on slot k of unit j within t
CL_{ijt}	Length of the current campaign of task i in unit j by the end of t

Integer

n_{ijt} number of batches produced in the current campaign of task i in unit j by the end of t

Δn_{ijt} number of batches of task i produced in unit j within t

Binary Variables

x_{ijkt} 1, if unit j runs a campaign of task i on slot k of t

u_{rijkt} 1, if resource r on slot k of t is used by unit j running a campaign of task i

z_{ijt} 1, if unit j runs a campaign of task i in t

$y_{ii'jkt}$ 1, if task i on slot k of unit j in interval t precedes i'

Chapter 7

Notation

Indices

i Task

l Production lines

m Materials

s Site

t Time interval

p Period

Superscripts

L Lower limit

U Upper limit

Sets

L_s	Lines that are at site *s*
I_l	Tasks that line *l* can perform
M_s	Materials that that site *s* either consumes or produces
IM_s	Materials that site *s* consume
OM_s	Materials that site *s* produce
PC_m	Set of all precursor materials for final product *m*
T_p	Intervals in period *p*
FP	Set of final products for E
PC_m	Set of all precursor materials of a final product *m* in E

Parameters

σ_{mi}	Mass ratio for material *m* in task *i*
$\delta_{mss'}$	Lead time for procuring material *m* at *s′* from *s*
A_m	Shelf-life of material *m*
h	Length of an interval *t*
τ_{ils}^U	Cycle time of *i* in *l* at *s* for minimum resource allocation
τ_{ils}^L	Cycle time of *i* in *l* at *s* for for maximum resource allocation
Q_{ms}^U	Storage capacity of material *m* at site *s*
B_{ils}	Batch size of task *i* on line *l* at site *s*
D_{mst}	Demand of material *m* at site *s* at time *t*
OSQ_{mt}	Overall safety stock limit for product *m* and its precursor material at time *t*
SQ_{mst}	Safety stock of material m at site *s* at time *t*
$TP_{mss'p}^U$	Lower limit on transfer price for *m* from *s* to *s′* during period *p*
$TP_{mss'p}^L$	Upper limit on transfer price for *m* from *s* to *s′* during period *p*
a_{ils}	Constant for processing cost

b_{ils} — Constant for processing cost

hc_{msp} — Cost of holding unit material *m* at site *s* during period *p*

γ_{msp} — Penalty for violating safety stock of *m* by unit amount at *s* during *p*

$D_{mss'p}$ — Duty for importing unit quantity of *m* from *s* to *s′* during *p*

tax_{sp} — Corporate tax at site *s* during period *p*

Dep_{sp} — Depreciation rate at site *s* for a period *p*

Variables

n_{ilst} — Number of batches of task *i* in line *l* at site *s* during *t*

cl_{ilst} — Length of a campaign of task *i* in *l* at *s* during *t*

Q_{mst} — Net usable stock of material *m* at site *s* at time *t*

$OQ_{mss't}$ — Amount of material *m* received from site *s* to site *s′* at time *t*

Q^{a}_{mst} — Net stock of material *m* with an age of *a* intervals at site *s* at time *t*

$OQ^{a}_{mss't}$ — Amount of *m* with an age of *a* intervals received at *s′* from *s* at time *t*

q^{a}_{mst} — Amount of *m* with an age of *a* intervals consumed at *s* during interval *t*

$\Delta OSQ_{mss't}$ — Amount of material *m* at time *t* violating overall safety stock limits

ΔSQ_{mst} — Amount of *m* at *s* violating site-specific safety stock limits at time *t*

$\Delta TP_{mss'p}$ — Differential transfer price over and above the minimum

PU_{lsp} — Total idle time of line *l* at site *s* during period *p*

R_{sp} — Revenue of site *s* during period *p*

IBT_{sp} — Taxable income of site *s* during period *p*

ATP_{sp} — After tax profit of site *s* during period *p*

NP — Total profit of E

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The pharmaceutical industry touches every human life on this planet. Arguably, the origin of pharmaceutical products may be dated back to the era of the inception of human civilization. Since then the industry has undergone a tremendous transformation from the scale of concocting herbs by a 'medicine man' to highly complex, sophisticated and large scale manufacturing facilities. The importance of the modern pharmaceutical industry is evident from the fact that three of the eight millennium development goals set by the UN [1], 'reducing child mortality', 'improving maternal health', and 'combating HIV/AIDS, malaria and other disease' depend upon improving access to medicines. Specifically, one of the Millennium Development Goal targets is, "in cooperation with pharmaceutical companies, (to) provide access to affordable essential drugs in developing countries". Given the importance, adequate supply of drugs at affordable prices is crucial. One of the key factors to ensure this is the effective and efficient operations of pharmaceutical industry.

Gone is the era, when pharmaceutical companies used to enjoy hefty returns from a few 'blockbuster' drugs and cared less for the development and production costs. The surge of 'me-too' (generic) drug companies and stringent regulatory protocols along with shrinking product pipelines and peaking patent cliffs has completely changed the way these companies are operated. In its quest for competitive advantage and sustainable profits, the industry has witnessed numerous mergers, acquisitions, and partnerships in the past couple of decades. Table 1.1 shows a glimpse of a few major mergers and acquisitions in pharmaceutical and healthcare industries for the past two decades. In addition to this, irreversible globalization, increasing environmental regulations, and new scientific advances have led to a noticeable operational reorganization including, drug discovery, clinical trials management, drug launch and marketing, production, warehousing and distribution, product tracking, and drug delivery mechanisms. This paradigm shift in the dynamics of global pharmaceutical business has thrown a slate of operational challenges for the sustainability of these companies. Thus, to meet such challenges and remain competitive in the market, companies seek cutting edge technologies and integration tools for operations management and resource handling. The objective is to minimize the operational/development costs, maximize profits, comply with environmental and regulatory protocols, and yet meet the societal needs. Figure 1.1 depicts the 'trilemma' of sustainable pharmaceutical companies.

Table 1.1 Major mergers and acquisitions in pharmaceutical and healthcare industries

Company	Target company	$ billion	Technology/product
Pfizer	Werner Lambert	90	Lipitor
Pfizer	Wyeth	68	Prevnar, Enbrel Pharmaceuticals
Sanofi Aventis (Sanofi)	Aventis	62	
Pfizer	Pharmacia	57	Celebrex
GSK (Glaxo Wellcome)	Smith Kline French	55	
Merck	Schering Plough	41	Pharmaceuticals
Astra	Zeneca	35	
Novartis (Ciba Geigy)	Sandoz	26	
Bayer	Schering	19.7	Pharmaceuticals
Schering Plough	Organon	14.5	Pharmaceuticals
Takeda	Nycomed	13.6	Pentaprazole, Daxas/Daliresp
Sankyo	Daiichi	7.7	Pharmaceuticals
Abbott	Solvay	7	Tricor, Trilipix, vaccines
Nycomed	Atlanta	6	Protonix
UCB	Schwartz	5.8	Pharmaceuticals
Teva	Ratiopharm	5	Generics
Daiichi Sankyo	Ranbaxy	4	Generics
Abbott	Kos	3.7	Humira, Niaspan
Abbott	Piramal	3.7	Generics
GSK	Steifel	3.6	Dermatology
Pfizer	King	3.6	Analgesics

Now, at the heart of all aforementioned objectives and issues lies a key question of how to optimally use the available resources and technologies in the presence of real and practical constraints. This is precisely a situation where optimal planning and scheduling of the supply chain operations have a huge and critical role to play. Although studies on planning and scheduling in the context of pharmaceutical industry exist since 1950s, they are still improving. Broadly, such studies in the open literature are classified under different stages of drug development and production such as product pipeline, clinical trials, primary and secondary manufacturing, warehousing, and distribution. To this end, this PhD research focuses on the planning/scheduling drug manufacturing, warehousing, and distribution operations. Specifically, this research aims at (a) identifying critical issues in the operational planning and scheduling in the context of pharmaceutical industry, (b) bridging gaps between industrial requirements and academic research by developing effective decision support models and tools, and (c) defining new frontiers for future research that eventually may improve the sustainability of pharmaceutical enterprises from all three perspectives, i.e. economic, social, and environmental.

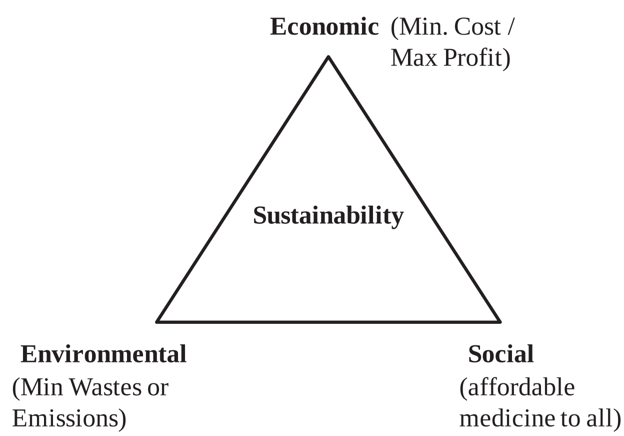

Figure 1.1 'Trilemma' of sustainable pharmaceutical companies

The following sections discuss more on global pharmaceutical industry, its supply chain structure, managerial hierarchy, and highlight the need and importance of better decision support models / tools for enhancing the sustainability of pharmaceutical companies.

1.1 Global Pharmaceutical Industry

Globalization and urbanization are posing unprecedented challenges and creating new opportunities for the pharmaceutical companies around the world. The impact of improved urban mobility and the level of preparedness in dealing with new diseases are evident from the recent experiences with SARS and H5N1. Also, environmental shifts are expected to transform pharmaceutical market. In one such scenario, scientists believe that global warming can bring [2] diseases such as malaria, cholera, and diphtheria to more developed countries. In addition, the overall demand is growing rapidly with the growing and aging population. According to a recent report [3], the global population is expected to be around 7.8 billion by the end of 2020 and 10 billion by the end of 2050. This population is also aging rapidly as by 2020 about 860.9 million people (11% of world's inhabitants) will be 65 or more, compared to 629.6 million people (9% of world's inhabitants) in 2010. On an average, older people consume more medicines than the younger people and thus, contributing to the total consumer base. Consequently, the global pharmaceutical market is expected to worth about $800 billion by 2020 [2].The major economic challenges include shortening of patent protection and the rise of generic companies. A recent report [4] states that over the next five years, products that currently generate more than $142 billion will lose patent protection and face generic competition, including Lipitor, Plavix, Zyprexa and Levaquin.

In order to meet the aforementioned challenges and to grab new opportunities, it is imperative for companies to effectively use their resources and minimize costs. In this regard, streamlining and optimization of the supply chain operations in manufacturing and distribution offer a huge potential for cost reduction. However, pharmaceutical companies are not known to be the best practitioners of efficient supply chain techniques. This is evident from ‘The Gartner Supply Chain Top 25’ report for 2011 [5], as only 1 pharmaceutical company (Johnson and Johnson) features in the top 25 and only 3 in the top 50.The adoption of best supply chain practices is partly hindered by the stringent regulatory protocols and confidentiality issues. A study [6] based on ten largest global pharmaceutical companies during 1996 – 2005 shows that firms spend around $699 billion on manufacturing. This, according to the same study, is nearly same as the expenditure on marketing and twice as the expenditure on R&D. Multiple studies [7, 8] estimate that the possible savings in the manufacturing of pharmaceutical products are in the range of $20 to $50 billion every year. Also, it is likely that manufacturing is the only sector that provides the opportunity for cost reduction. This is because, the companies may not be willing to reduce the expenditure on R&D and are also unable to cut marketing costs due to a fierce market competition.

1.2 Pharmaceutical Supply Chain

The pharmaceutical supply chain is responsible to ensure that the right drug reaches the right person, at the right time and in right conditions. In addition, it is highly critical – perhaps not surprisingly – given that it deals directly with the health and safety. So, anything less than a 100% customer satisfaction is inadmissible. Thus, not surprisingly, pharmaceutical supply chain is highly sensitive and regulated. The supply chain problem in pharmaceutical industry consists mainly of two parts, drug development and drug production. The supply chain concerning the identification, testing and then getting to the level of commercial production is an important problem. However, in this research, we focus on the supply chain concerning the manufacturing and distribution of the commercial and new products. The unique characteristics of pharmaceutical companies such as long cleaning/set-up times, resource intensive operations, multi-step synthesis, short material shelf-lives, and high waste generation make its supply chain different from that of other industries. Also, the pharmaceutical supply chain faces several unique challenges such as parallel trade and drug counterfeiting, visibility across an extended supply chain (including that of external suppliers and distributors), and pricing. In the past, pharmaceutical companies depended heavily on the introduction of new products, increasing demand, and maintaining high product availability. This prevented the supply chain efficiency from being the biggest challenges of the industry. However, as the companies are now striving to reduce costs and improve efficiency, supply chain appears to be highly promising and interesting.

Typically, the supply chain network of a global pharmaceutical company is extremely complex and involves a number of entities. It extends from the lab-scale testing and synthesis of a drug to multiple tiers of clinical trials and then to the industrial scale production and distribution. In this research project, we focus on a part of this supply chain consisting of production and distribution. Figure 1.2 shows a typical configuration for this part of pharmaceutical supply chain. It consists of multiple production and distribution facilities, raw material suppliers, and customers located around the world. First, the primary manufacturing facilities procure raw materials from different suppliers and convert them into active pharmaceutical ingredients (APIs) or 'white powder'. APIs are the main ingredients in a drug with medicinal/pharmaceutical properties and are responsible for the diagnosis, cure, treatment, or prevention of diseases. These APIs are then used by the secondary manufacturing facilities along with excipient materials to formulate drugs in a specific form including pills, tubes, tonics or gels. Excipients are the pharmaceutically inert materials that act as the carrier for APIs. The final drugs are sent to various distribution facilities in bulk quantities for packaging and labelling according to their final market regions. Finally, the packaged drugs are sent to the customers from different market regions. In addition, each step of manufacturing and distribution engages a number of resources making supply chain operation resource intensive.

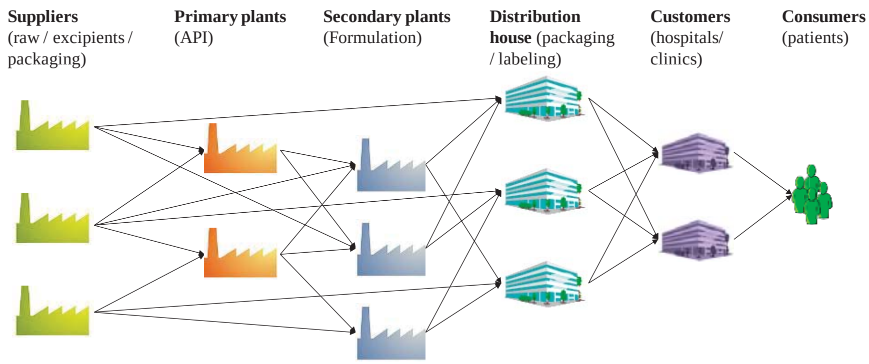

Figure 1.2 A configuration of a typical pharmaceutical enterprise

1.3 Hierarchical Decision Making

Now, the aforementioned objectives such as enhancing supply chain efficiency and reducing production costs essentially require optimal resource utilizations, minimum losses, and efficient operations. In other words, improving supply chain efficiency and reducing costs require efficient planning and scheduling of the operations and resources. In this regard, given the inherent complexity of supply chain operations and decision making, intelligent and sophisticated tools for decision support are very useful.

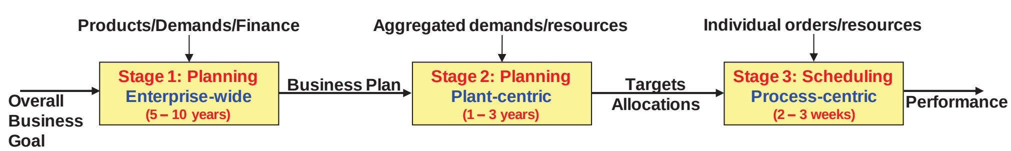

Figure 1.3 Decision making hierarchy in a typical pharmaceutical company

Typically, managerial hierarchy for decision making in a typical pharmaceutical industry (Figure 1.3) can be broadly categorized into three levels: enterprise –, plant –, and process–centric. At the enterprise level, a higher management team prepares a business plan involving long-term strategic decisions for around 5-10 years. This business plan mainly focuses on the overall performance of the enterprise and sets targets for each SBU (strategic business unit). Specifically, the decisions involved here include resource allocations such as capital budgets and product portfolios, capacity expansion, outsourcing of processes or services, manpower planning, number of NPIs (new product introductions), profit margins, etc. Thus, the business plan involves aggregate objectives that are focused on the overall profitability of the enterprise as a whole and usually does not involve any finer details. The plant centric management team uses this business plan as a basis to draw their short-term plan for 1-2 years. This short-term plan involves annual/bi-annual targets that are specific to a plant. This include resource allocation within the plant such as purchasing new equipment, NPIs, production targets, manpower allocation, maintenance planning, etc. Here, finer details of the various processes and unit-operations are usually not considered. The process owners or the production managers consider these aggregate targets as a basis and prepare the plant production schedule. The production schedules are usually meant for a few weeks or a couple of months and include more specific and finer details of the process. The production schedules usually consider all process attributes such as batch sizes, processing times, campaign lengths, changeover or setup times, cleaning times, product inventories, recycling, etc.

1.4 Need for Planning and Scheduling

Given a hierarchical decision making structure in a pharmaceutical company, the challenges involved at each level of hierarchy are different. For instance, planning activity requires collaboration among the different departments for a number of inputs such as demands, resource availabilities, inventories, maintenance, and NPIs.

Sometimes, finding a feasible plan that addresses the concerns and needs of all stakeholders involved is in itself a very complex problem, leave aside finding an optimal plan. Usually, the planning office is responsible for operational planning and scheduling and for communicating with all stakeholders. This along with the complex chemistry involved in the drug manufacturing process makes planning and scheduling a highly complex endeavour. Specially, scheduling involves a varied set of decisions such as allocation of different product stages to the production lines, batch numbers and sizes, campaign lengths, cleaning and set-ups, material transfers, storage, etc. Given the myriad combinations in which such decisions can be made, identifying the optimal set of decisions is very difficult. The scale and size of the problem makes the problem very difficult for a human being to manually find the best plan or schedule for a given scenario. At the enterprise level, planning involves integrated decisions for procurement, production, and distribution. Specifically, for a global pharmaceutical company, planning includes a set of constraints such as raw material procurement, production planning, inventory management, waste handling, material shelf-life tracking, material transfers and lead times, and impacts of international tax-differentials. Again, these decisions are usually for a long horizon and involve uncertainties. This makes the problem very difficult to solve. Clearly, intelligent tools and advanced models are necessary to help the decision makers in finding good solutions.

Consequently, the problem of production planning and scheduling in pharmaceutical industry has attracted a lot of interest from the industrial and academic research communities. As shown in chapter 2, a remarkable progress has been made since the inception of mathematical modelling techniques (linear programming, integer programming, etc.). However, it is also clear from the literature review that the existing models and technologies are still improving. Thus, further work is required to achieve better and efficient models, techniques, and tools. The main challenges are in dealing with computational tractability, industrial acceptability, and combinatorial complexity of the planning problem.

1.5 Research Objectives

This research primarily focuses on developing advanced mathematical models and decision-support tools for planning and scheduling in pharmaceutical industry. The problem of planning and scheduling is studied extensively in the open literature. However, much of this study is centric to general batch/semi-continuous plants and thus, lacks specific considerations of pharmaceutical companies. Also, the existing generic models are limited by their approach and assumptions. In principle, real-life scheduling problems have myriad of considerations and it is very difficult to formulate a full-scale scheduling model to conceive an optimal solution. Even if a comprehensive scheduling model is constructed, it is nearly impossible to evaluate all possible alternatives in a simple manner to find an optimal solution (combinatorial complexity). The challenge is to have efficient models that can give quick and good solutions that are both scalable and closer to the real-life problems. The specific objectives of this research are, therefore, to (1) analyse the existing 'best' scheduling models, identify and address their limitations, and explore new approaches for short-term scheduling of generic batch plants, (2) extend considerations of the current scheduling models to push them a little closer to their realistic counterparts and develop new and/or improve existing models to efficiently solve short-term scheduling problems, (3) develop integrated frameworks and methodologies to study the dynamics of multi-site enterprise planning and the effects of resource allocation on production planning (4) identify industrial requirements and develop intelligent decision-support tools that are scalable, robust, and give industrially acceptable solutions.

1.6 Outline of Thesis

This thesis consists of nine chapters. After a brief introduction in Chapter 1, a detailed literature review discussing existing approaches and models for planning and scheduling batch plants in general and pharmaceutical plants in particular is presented in Chapter 2. A number of gaps in the literature and the directions for future work are then identified and summarized.

In Chapter 3, some recent unit-specific event-points based scheduling models are analysed. Their limitations and suggestions to address such limitations are discussed. It is shown that by not addressing these limitations, some unit-specific event-based models may lead to non-optimal solutions in some cases. Three examples involving shared and limited storage are presented to demonstrate our findings.

In Chapter 4, with the motivation of findings in Chapter 3, a novel approach to scheduling multipurpose batch plants using unit-slots instead of process-slots to manage shared resources such as material storage is presented. Here, two slightly different but compact and simple models are developed. This multi-grid approach rationalizes, generalizes, and improves the current multi-grid approaches for scheduling with shared resources. Also, the models allow non-simultaneous transfers of materials into and out of a batch, which is shown to give better schedules than those from existing models in some cases. Furthermore, the presented approach requires fewer slots (event-points) on some examples than those required by the unit-specific event-based models.

Chapter 5 extends and generalizes the multi-grid scheduling approach based on unit-slots presented in Chapter 4 and the single-grid approach from literature based on process-slots to consider rigorous resource constraints. Here, a number of real-life scheduling considerations such as sequence-dependent set-ups, effects of resources (other than material and equipment) on scheduling, non-simultaneous material transfers, non-zero transfer times, and multiple storage configurations are incorporated. In addition, different variations for the presented models that appropriately suite their application to a given problem are discussed.

In Chapter 6, the concept of resource availabilities affecting production scheduling (Chapter 5) is generalized and studied with a strategic perspective. A framework is developed to study the effect of resource allocation on the process performance. Also, a few key aspects of the industrial planning activity such as interactions among the planner and other stakeholders, campaign mode operations, and safety stock policy are considered. A simple mathematical model for integrated resource allocation and campaign planning is presented. The model enables decision support pertaining to campaign scheduling, sequence-dependent changeovers, key resource allocations, scheduled maintenance, inventory profiles with safety stock limitations, and new product introductions.

Chapter 7 extends the integrated problem of resource allocation and campaign planning of Chapter 6 from a single plant and considers the entire production supply chain of a multinational pharmaceutical enterprise. A simple yet powerful model for multi-period enterprise-wide planning is presented. Here, the entire enterprise is represented in a seamless fashion with a granularity of individual task campaigns on each production line. The model considers an integrated problem of procurement, production, and distribution and incorporates several practical features of industrial planning such as effects of international tax differentials, inventory holding costs, material shelf-lives, waste treatment / disposal, and other real-life factors on the after-tax profit of a company.

In Chapter 8, a tangible outcome of this research in the form of a decision-support tool 'PlanPerfect', for integrated production planning and resource allocation for pharmaceutical plants is presented. The tool is developed in association with a Singapore-based plant of a multinational pharmaceutical company. PlanPerfect is motivated from the complex problem of production planning existing at the associated plant. It is specifically designed and customized to address the needs of planners in any pharmaceutical plant.

Finally, conclusions and recommendations for future research are summarized in Chapter 9.

2 LITERATURE REVIEW

In general, pharmaceutical plants are operated in batch mode and produce a variety of products through multi-step / multi-stage processes using multipurpose equipment and resources. The production planning and scheduling problems for batch plants are being studied since the introduction of linear programming and integer programming techniques in 1950s [9-12]. In the last three decades, a significant progress has been made in this field of research. A number of models and approaches have been developed precisely to address this problem of high importance. However, the production planning and scheduling problems in the specific context of pharmaceutical plants have received a little attention. Also, the existing models and approaches for planning and scheduling batch plants are still being improved for faster computing and solving larger problems [13-15]. This highlights the need of more work and inspires the direction of such work in future. To this end, this chapter is organized as follows. First, the best approaches from the literature are discussed and the existing models are reviewed in the context of planning and scheduling batch plants in general and pharmaceutical plants in particular. Next, a set of gaps is identified, challenges are discussed, and scope of this research is stated.

2.1 Approaches for planning and scheduling

Research efforts in the last few decades have resulted in a great variety of approaches for scheduling and planning batch plants. In this study, we broadly classify the existing approaches as black-box or detailed mathematical modelling approaches, based on the level of details that these approaches require in modelling a planning or a scheduling problem.

Black-box modelling approaches use meta-heuristic methods or evolutionary algorithms such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), particle swarm optimization (PSO), and ant-colony optimization (ACO). These methods essentially make a few or no assumptions for the problem and search for the best solutions in a set of candidate solutions. Although such meta-heuristics approaches neither guarantee nor can prove optimality, a general consensus and the extensive work in literature shows that they can provide optimal or near-optimal solutions within moderate computation power. He and Hui have used several variations of genetic algorithm in a series of publications [16-20] for scheduling batch plants. Their main objective in the articles is to solve large-scale problems. Jou [21] presented a special algorithm based on GA for a production scheduling problem. Azzaro-Pantel et al [22] presented a genetic algorithm based bi-level solution strategy for scheduling batch-plants. Ku and Karimi [23] presented a simulated annealing framework for scheduling batch plants with unlimited intermediate storage. They further presented a comparison with three heuristic methods for the objective of make-span minimization. Raaymakers and Hoogeveen [24] presented a simulation annealing based model for scheduling multipurpose batch plants with no-wait policy for some materials. Patel et al. [25] used the methodology of simulated annealing for the design of multiproduct non-continuous plants. They presented a strategy to choose the best set of annealing parameters and evaluate several forms of simulated annealing algorithms. Shelokar et al. [26], Jayaraman et al. [27], and Heinonen and Pettersson [28] used ant colony classifier based optimization algorithm for scheduling different process systems. Liu et al. [29] presented a hybrid particle swarm optimization technique for batch scheduling. They used a case study from polypropylene production plant to develop the algorithm.

The other highly studied approach is based on detailed mathematical modelling. This approach offers guarantee for optimality but sometimes is computationally expensive. Numerous models are developed in the past using this approach. The most important issue that differentiate these models is the representation of time. Based on the assumption of whether the process operations in a schedule have to begin at some pre-defined time points or may begin at any point within the horizon, all these models can be broadly classified into discrete- or continuous-time models. Discrete-time formulations can be further classified into uniform and non-uniform discrete time model. In the former, the horizon is divided into a finite number of intervals of equal and known duration. In the later, the durations are known but are not uniform across the horizon. A number of formulations [30-40] following both these approaches exist in the literature. As process operations are forced to begin or end at the boundaries of these discrete intervals, the resource balance is very straight forward. However, such approaches may easily become computationally intractable as the number of intervals increase and often may not lead to the optimal solutions because of a priori discretization of time. Thus, continuous-time formulations have gained popularity in the recent past as they do not have such limitations. Also, in this research work, we focus on studying the existing and developing new formulations based on continuous-time approach. In the literature, continuous time formulations have been classified into slot-based, event-based, and precedence-based (or sequence-based) models based on the details of time representation in the model.

The sequence- or precedence-based representation [41] employs either direct- [42-46] (immediate precedence) or indirect- [47-50] (general precedence) sequencing of pairs of tasks on units. While it does not use constructs like "slots" or "event points" in time explicitly, it does assign times for various tasks and must pre-postulate the possible numbers of batches. The handling of shared resources is not straightforward with these models.

Wagner [51] defined "order-positions" in a task-sequence for a machine-scheduling problem. He assigned one task to each position and one position for each task. Later, Ku and Karimi [52-54] and Birewar and Grossmann [55] used "positions in a sequence" or "production slots" for scheduling multiproduct batch plants. These "positions" are essentially slots, and these models are slot-based. Their slots are in a sequence (or sequence-slots) rather than time (time-slots). These models restricted each task to only one sequence-slot, and did not use any explicit time-grid. However, they assigned timing variables such as completion times to these sequence-slots, which mapped time-slots on the time-grids of various units.

Apart from the aforementioned models, all slot-based scheduling models that we are aware of define slots as time intervals on a time-grid, or simply as time-slots. Geoffrion and Graves [56] divided the scheduling horizon into "equal indivisible known time slots" (uniform discrete-time representation). In contrast, Sahinidis and Grossmann [57] used non-identical, ordered time-slots of unknown variable lengths in their continuous-time formulation for planning continuous processes. More significantly, they appear to be the first to allow a task to span multiple consecutive time-slots, thus generalizing the assumption (one slot per task) used by the earlier sequence-slots based models [51-54, 58]. Since then, several formulations [13, 57, 59-71] have employed slots as ordered time intervals of unknown and variable lengths rather than sequence positions. Although some researchers [46, 63, 65-67] restricted each task to a single slot, others [13, 57, 59-62, 68-71] have allowed it to span multiple slots. In our view, *all* are slot-based models with the former being just a special case of the latter. This is indeed the traditional view on slot-based models since the work of Sahinidis and Grossman [57].

The scheduling literature has used two types of time slots. One is the synchronous [68, 70, 71] or process-slots [46], and the other is asynchronous [63] or unit-slots [46]. In the former, all units in a process share one common set of slots. This provides a single time-axis to balance shared resources such as storage, utilities, and manpower with ease. In the latter, each unit has a separate (or unit-specific) set of slots with partially or wholly independent timings. Since the slot timings vary independently with units and are unknown in a scheduling formulation, the "order" of any resource usage during the slots is not readily known on a single time-axis. This makes the resource balance difficult.

Zentner et al. [39] used terms such as "events" and "event times" for the starts/ends of tasks in their comparison of uniform discrete-time models and non-uniform continuous-time models. Zhang and Sargent [72, 73] used the same concept of "events" in their continuous-time MINLP formulation for an operational planning problem. These "events" can be viewed [12] as "starts/ends" of slots. Ierapetritou and Floudas [74] introduced the concept of "event points" in their continuous-time model. Floudas and Lin [12] classified these into global and unit-specific event points. In the former [75, 76], each event point has a unique value of time, which is the same for all units. This orders the event points on a single time-axis. This is the same [76] as synchronous or process-slots, because the interval between successive event points is nothing but a process-slot. In unit-specific event points, an event point [74] is associated with multiple time instances, one for each unit (or unit-specific). Tasks corresponding to the same event point start at different times on different units. This is the same as unit-slots, because the interval between the timings of two successive event points on a unit is a unit-slot. Thus, both event-based and slot-based approaches are conceptually the same, as they use time grids. While the former views grid/s in terms of ordered, distributed time points, the latter does the same in terms of ordered, variable-length time intervals.

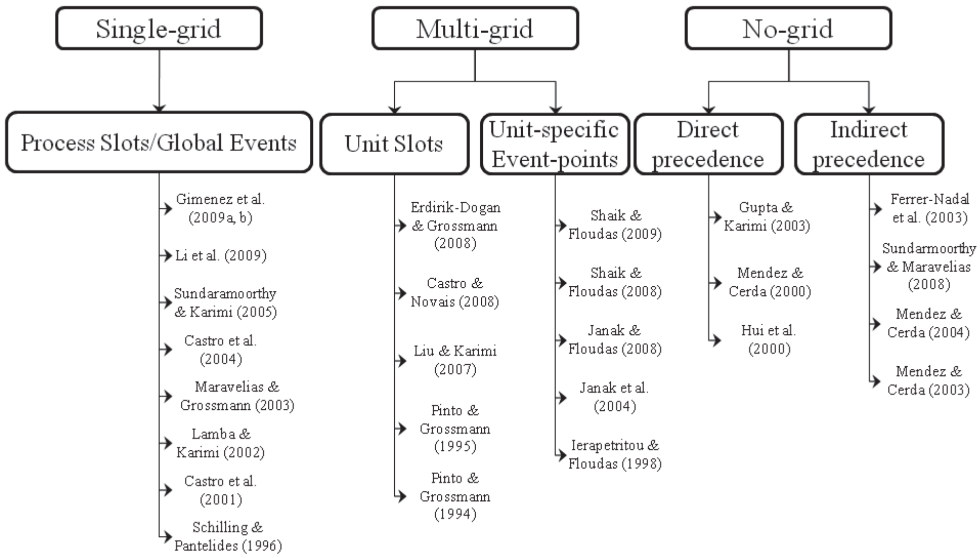

Figure 2.1 A classification of continuous-time scheduling models

Given the conceptual resemblance between the slot-based and event-based models, the notion of time-grid [77, 78] seems clearer for classifying various scheduling approaches. Using that notion, we suggest three types of models (Figure 2.1), namely single-grid, multi-grid, and no-grid. In order to define a schedule, task timings are highly important. So, all approaches are required to use timing variables (in different forms) for the occurrence of various activities. However, the sequence-based approach does not employ a "grid" defined in terms of either "event points" or "slots". Thus, no-grid does not mean that timing variables do not exist. Instead, it just signifies that the timing variables are not associated with a grid. In that sense, one could argue it to be a no-grid approach. The models using global events or process-slots employ one common time-grid for all units and resources and thus, they are single-grid models. The models using unit-specific events or unit-slots employ a separate time-grid for one or more units or resources and so, they are multi-grid models [77].

While the above classification presents our viewpoint, several different interpretations are indeed possible. In addition, some remarks are in order. While the single-grid models do not differ significantly in modelling details, some multi-grid models do, especially in terms of variables and constraints. In this regard, we wish to highlight some differences between two groups of multi-grid models. One group has used the so-called unit-slots [46, 59, 63, 67] and the other has used the so-called unit-specific events [79-82]. First, only a few models from the former have addressed shared resources. Karimi and McDonald [60] used a mix of unit-slots and non-uniform periods (process-slots), and restricted inventory balances across periods only. Lim and Karimi [63] and Erdirik-Dogan and Grossmann [59] used unit-slots, but both defined binary variables to relate the timings of resource usage across various units. In contrast, the unit-specific event-based models have handled shared resources without using any such binary variables. Second, most unit-specific event-based models define task timings as 3-index (task i, unit j, event point n) variables with unit-index being explicit [83] or implicit [74, 77, 81, 82]. In contrast, the models using unit-slots invariably define timings as 2-index (unit j, slot k) variables with no index for task. This difference makes the model details (variables and constraints) and characters different for these two groups of models. Because tasks generally outnumber units, the event-based models need more event-time variables ($i*j*k$ vs. $j*k$) and associated timing constraints.

2.2 Scheduling Pharmaceutical/Batch Plants

The flexibility and versatility of batch plants in general provide both opportunities and challenges for the manufacturer. Méndez et al. [84], Floudas and Lin [12], and Pitty and Karimi [41] presented excellent reviews of the current approaches and associated challenges for the short-term scheduling of the batch processes. The main considerations in most scheduling problems are the allocation of equipment and resources to various tasks and sequencing them over time. In the literature, scheduling pharmaceutical plants or batch plants is broadly studied under two different operating conditions, batch scheduling and campaign scheduling. Another important feature is addressing a scheduling problem in the presence of uncertainties in process parameters. Although there are a number of ways of dealing with uncertainties in the literature, we found that the approach of reactive scheduling is more appropriate in this research project.

2.2.1 Batch Scheduling

Batch scheduling essentially involves batch to batch transitions, complex network flows, batch splitting and mixing, etc. Several formulations of different types exist in the literature for scheduling batch plants. Karimi and McDonald [60] used a combination of multi-grid (unit slots) and single-grid (process slots) approaches and presented a multi-period scheduling model for semi-continuous processes. Ierapetritou and Floudas [74] proposed a multi-grid model based on unit-specific event points for the scheduling of multipurpose batch plants. Maravelias & Grossmann [76] and Sundaramoorthy & Karimi [70] have presented similar models based on single-grid approach for scheduling multipurpose batch plants. They, however, have significant modelling differences as detailed by Sundaramoorthy & Karimi [70]. Unlike Sundaramoorthy & Karimi [70], Maravelias & Grossmann [76] presented a few additional constraints for sequence-dependent changeovers and utility consumption. For several examples, it was shown that the former model requires fewer slots / events and so, is faster than the later. However, the excessive number of binary variables required by the single-grid models in general limits their application to large-scale problems. Janak et al. [81] proposed a MILP formulation that improved over the previous versions of the unit-specific event-based models. They defined additional binary variables for denoting task ends. They also introduced storage tasks to address various storage configurations and even allowed a task to span multiple event-points, which was a significant departure from the earlier unit-specific event-based models. However, this formulation required many constraints and variables, and results in excessive solutions times [82]. Shaik & Floudas [78] and Shaik et al. [85] proposed an improved version of the model of originally presented by Ierapetritou and Floudas [74]. However, it was later shown [82, 86] that these improved models may not be able to yield optimal solutions for some problems. Addressing this, Shaik & Floudas [82] improved the model of Janak et al. [81] using a 3-index binary variable and without using storage tasks. Their model uses a user-defined parameter to allow tasks to span a given number of event-points. However, it requires one to iterate on that parameter for every possible number of event-points to reach an optimal solution. Also, the additional set of iterations on this parameter may confuse users in identifying the optimal solution. Castro & Novais [83] presented a multi-grid model for scheduling multi-stage and multiproduct batch plants with unlimited intermediate storage using a resource-task-network (RTN) approach. Castro et al. [77] presented a simultaneous design and scheduling model for MBPs using single-grid approach, which is an extension of their earlier model [75]. They allowed non-simultaneous transfers of materials. For this, they defined additional transfer tasks and constraints to allocate material transfers to such tasks. They do not allow temporary storage of materials in the processing unit. Mendez and Cerda [48] and Ferrer-Nadal et al. [50] have proposed sequence-based (no-grid) models based on indirect-pairing (general-precedence) of tasks. Ferrer-Nadal et al. [50] addressed non-zero transfer times from processing to storage units and vice-versa. While these two models do not need to pre-postulate the numbers of event points or slots, as mentioned earlier, they assume the number of batches for each task.

2.2.2 Campaign Scheduling

Scheduling in particular to the pharmaceutical industry involves accounting for huge cleaning and set-up times along with the product changeovers. To avoid this huge cleaning times, the plants are generally operated in long product campaigns. However, long campaigns result in high inventory levels, which again incur inventory holding cost. Thus, campaign mode operation involves a trade-off between the flexibility availed through batch operations and the inventory costs.

Mauderli and Rippin [87] initially pointed out the importance of campaign mode operational strategy in batch process production scheduling. Outlining the requirements of a computer program for scheduling multipurpose batch processes, they emphasized the need of such techniques for the ease of management decision making. They developed a LP based screening procedure to identify a set of dominant campaigns and then used a MILP formulation for scheduling those campaigns. Birewar and Grossmann [58] addressed the problem of campaign scheduling at the design stage. They developed a NLP model for scheduling and designing multi-stage and multi-product batch plants and assumed UIS/ZW storage policy. Ku and Karimi [52-54] proposed sequential slot based mathematical formulation for batch scheduling with various storage policies and also, extended to the case of intermediate due dates. Wellons and Reklaitis [88-90] proposed a MINLP formulation for the scheduling of single product campaigns with NIS/ZW policy. They further proposed a decomposition strategy to reduce the computational effort and to avoid the degeneracy resulted through the model. Shah and Pantelides [91] developed a MILP formulation for the campaign scheduling of multipurpose batch plants. Their formulation was compatible with the limited resource availabilities, which was a significant departure from the previous approaches. Papageorgiou and Pantelides [92] proposed a three level sequential approach for campaigning multipurpose batch plants. The three levels of their approach broadly identify suitable equipment allocations; batch sizes/campaign lengths, and appropriate timings. Voudouris and Grossmann [93] extended the work of Birewar and Grossmann [58] to address campaign planning in multiproduct batch plants using cyclic scheduling, various inventory constraints. Also, they introduced several linearization schemes to reduce the original MINLP model to the MILP model. Tsirukis et al. [94] explicitly considered the resource constraints and developed a MINLP formulation. To reduce the complexity, they further decomposed the problem into two MINLP formulations, one to assign the orders to the campaigns and then to allocate resources to the tasks. Papageorgiou and Pantelides [95] addressed material recycling, shared resources, various storage policies, inventory holding costs, and cyclic scheduling in a multipurpose batch plant scheduling. However, they considered fixed campaign lengths, which increase the problem size with uniform discretization of the time horizons.

Later Karimi and McDonald [60] developed two efficient continuous-time slot-based MILP formulations for scheduling product campaigns in single-stage multiproduct batch facilities. They used the concept of minimum run length and several other details such as inventory management, outages, and maintenance. Lamba and Karimi [62] developed a novel MILP formulation using multi-grid time slots for the campaign scheduling of batch plants. They introduced extra binary variables to know the relative timings of each task. Rajaram and Karmakar [96] studied the campaign scheduling problem with its application to the food-processing industry. They incorporated several industry specific heuristics and used fixed cycle times to develop a MILP formulation for the problem.

2.2.3 Reactive Scheduling

Although scheduling only deals with a short-term optimization of the resource allocation and task precedence in a batch chemical plant, it is not free from the operational uncertainties such as process break downs, demand fluctuations, price volatility, etc. Particularly, in the case of pharmaceutical industry, where the products have high rates of return per unit sold, these unanticipated disturbances play a vital role in the overall profit. Thus, the reactive scheduling has received much attention in the literature. Reactive scheduling absorbs these unanticipated disturbances into the master schedule by making relevant adjustments to it over a short period of time. A more conventional approach would be to carry out a full scale rescheduling of the entire plant at every point of disturbance. However, this would be a costly affair and will result into a disruptive and non-smooth operation of the plant.

One of the initial efforts in this field was by Cott and Macchietto [97], who proposed an earliest finishing unit (EFU) heuristic in a shifting algorithm as a part of the bigger problem of production management of batch processes. Their algorithm allows the shifting of the start times of the various tasks on the affected units. However, the application was validated on small and medium sized problems with no general measure for its optimality or account for its profitability. Hasebe et al. [98] proposed a reactive scheduling algorithm for the multiproduct sequential batch plants. While their algorithm allowed insertion of a task or swapping of two tasks, they also mentioned that the generic approach of a full scale reordering of all the tasks simultaneously would be computationally demanding. Their algorithm worked with the aggregation and disaggregation of the similar tasks to reduce the problem size. Kanakamedala et al. [99] developed a reactive scheduling algorithm for the multipurpose batch plants. Their algorithm considers deviations related to the processing times and unit availabilities. Their approach was based on a least impact heuristic that tries to reduce the number of modifications to the master schedule. Huercio et al. [100] addressed a similar problem through time shifting of the tasks and unit reallocations. Their method generates a decision tree of alternative unit assignments for every deviation that is encountered. The best solution is then searched through a set of heuristic rules. Sanmarti et al. [101] extended the model of Huercio et al. [100] and showed the effect of considering robust schedules by generating a large number of failure sets and the consequent reliability statistics. They also included the cases of unanticipated equipment failure in their reactive scheduling model for multipurpose batch plants. Rodrigues et al. [102] extended the discrete-time scheduling model of Kondili et al. [34] incorporating reactive scheduling with uncertain processing times. They proposed rolling horizon approach for the same that also anticipates the possible future violations by calculating the criticality of processing time for each unit. Honkomp et al. [103] used Monte Carlo simulations for the rescheduling and performance evaluations. They develop two slightly different approaches based on uniform and non-uniform discretization of time. They imposed two main penalty functions for constant batch sizes and minimize variations to the master schedule. They further made use of some heuristic rules to identify the tasks that are more susceptible to violations. Elkamel and Mohindra [104] proposed a rolling horizon based decomposition strategy for accommodating the unanticipated disruptions into the master schedule of multipurpose batch plants.

Vin and Ierapetritou [105] extended the continuous time scheduling formulation of Ierapetritou and Floudas [74] to incorporate the rescheduling algorithm for disruptions in order timings and sudden breakdowns. Also, in order to reduce the computational effort they proposed to fix the variables corresponding to tasks prior to the unexpected events. Roslöf et al. [106] presented a heuristic reordering algorithm based on MILP, which performed priority based rescheduling of jobs in unanticipated situations. In case of unexpected process parameters, the algorithm is solved until either a predefined number of iterations is met or a convergence criterion is attained. Ruiz et al. [107] developed a fault diagnosis system for the multipurpose batch plants using artificial neural networks along with an expert system based on the knowledge of previous batches and plant model. The system when detects any process abnormality, it interacts with the optimizer to perform reactive scheduling. Méndez and Cerdá [48, 49] performed rescheduling for the multiproduct batch plants. They use a set of pre-ordering rules to reduce the size of the resulting MILP formulation in case abnormalities such as shifting of start times, new order arrivals, unit reallocation of both the old and inserted new batches. Later, they extended the model [49] to limited and discrete but renewable resources, where insertion of new batches was not allowed. Janak et al. [108] proposed a rolling horizon decomposition approach for the rescheduling of the batch plants based on several scenario-based heuristics. They considered two unexpected scenarios of equipment breakdown/shutdown and new or modified orders. For the implementation of their approach and problem size reduction they fixed the variables corresponding to the previous sub-horizons and performed full scale rescheduling of the remaining horizons. Recently, Ferrer-Nadal et al. [109] made use of a flexible recipe concept originally introduced by Rijnsdorp [110] to perform the rescheduling of multipurpose batch plants based on general precedence of tasks. They addressed the problem with non-zero transfer times, maintenance tasks, arrival delays, material shortages, quality concerns, new orders, due-date changes, and equipment failures.

2.3 Planning in the Pharmaceutical Industry

Planning is similar to scheduling and involve allocation of equipment and resources to various processes or operations and sequencing them over time. However, key differences between planning and scheduling are the time horizon under consideration, extent of process or operational details involved, and the level of uncertainties. Typically, planning involve decisions pertaining to campaign/batch scheduling, key resource allocations, new product introductions, maintenance, inventory management, material transfers, and changeovers in an integrated manner for a plant or an entire enterprise. Usually, planning is for longer time horizons as compared to scheduling and is of the order of 1 – 5 years. In a sense, planning gives a more strategic perspective to the operations of a company. In principle, a scheduling model can be used for planning. However, given the complexity of the scheduling problem, solving it for such long horizons will easily make this problem computationally intractable. Also, in most of the scenarios, the finer details of a scheduling model (e.g. start/end time of a task, sequence of batches, etc.) are not important in planning. Thus, there have been considerable efforts in the literature to develop efficient planning models. Also, efforts have been made to develop intelligent solution strategies that integrate planning decisions with scheduling models. Next, we review important contributions and efforts made so far for planning with a focus on the pharmaceutical industry and its operations.

2.3.1 Integrated production planning and scheduling

Typically, pharmaceutical industry strives hard to meet the flexible demands, which forces them to operate close to their capacity. So, it is imperative to have production targets that are feasible at the shop floor. In this regard, it is necessary to integrate the planning methods with the scheduling models. This method has its own pros and cons. Maravelias and Sung [15] presented an overview and consequently highlighted the limitations and opportunities of the existing methodologies for the integration of medium-term planning and scheduling in the chemical process industries. They briefly outlined the various integration schemes and methodologies that are prevalent in the literature. Also, they classified the existing integration approaches broadly into three main categories: hierarchical, iterative and full-space methods. They identified the generation of good integer cuts as a crucial step for faster convergence. Furthermore, the study pointed out several challenges such as the limitation of current approaches in solving the complex and large scale processes, the deterioration of the solution speed with increase in the number of iterations, exploring the advances in optimization solvers and other decomposition strategies.

In principle, the integrated problem of planning and scheduling can be solved with the detailed scheduling formulation for the whole planning period. However, this results into large problem size, which in-turn becomes computationally demanding. One of the much studied approaches for the problem size reduction has been the aggregation technique. Wilkinson et al. [111] proposed an aggregation approach in a multi-period scheduling problem. They aggregated small scheduling horizons to longer periods by keeping the disaggregated variables and constraints only near the boundaries of these longer periods. Since then, several aggregation techniques have been studied in the literature. Keeping the allocation variables and aggregating the timing constraints and variables has been one of the common techniques of aggregation approach [112-116]. The resulting aggregated model can be solved using various decomposition schemes. In decomposition methods, the main problem is split into two or more interacting ‘master-slave’ sub-problems. If the interaction is only in one direction i.e. form master (planning) problem to slave (scheduling) problem, it has been referred to as the hierarchical method or if there is a feedback loop from the later to the former it has been referred to as iterative approach [15]. Generally, the planning level aggregated problem is solved to obtain the decisions such as aggregate productions targets and also in some cases the allocation variables [117]. These aggregated targets are then supplied to the scheduling level problems to obtain the detailed schedules [118]. However, it may so happen that the scheduling problem turns out to be infeasible with the supplied production target from the upper-level planning problem. Several approaches [119] or heuristics [31, 120] exist in the literature to address the infeasibility at the scheduling level or realistic predictions at the planning level. Dimitriadis et al. [121] introduced an approach of rolling horizons algorithm, where the total planning period was divided into sub-periods and each period was scheduled using a lower-level scheduling model. This iterative approach uses detailed scheduling models for the earlier time windows (sub-periods) and the aggregated planning models for the later. The target for the later periods is updated as the horizon for scheduling model rolls on to the consecutive time windows. Since then several researchers have [103, 122-124] demonstrated the application of rolling horizon approach on batch scheduling problems.

Another approach of solving the decomposed problem is with a feedback loop from the scheduling level model to the planning model, i.e. iterative approach. Papageorgiou and Pantelides [125] developed a bi-level decomposition scheme with integer cuts to reduce the burden of some binary variables in the subsequent iterations. Stefansson et al. [126] developed a three-level hierarchical model with iterative approach for the campaign planning and scheduling in the secondary pharmaceutical industry. They have also reviewed the current approaches and called for further work regarding the dynamic procedures in the integration of planning and scheduling in process industries.

The other approaches of decomposition that have been widely used in the literature are the standard mathematical decomposition techniques such as Bender's decomposition [127, 128] and Lagrangian relaxation/ decomposition [129-133].

2.3.2 Production planning and resource allocation

Production planning is crucial and frequent in pharmaceutical companies. The complex and combinatorial nature of operations in which many products and intermediates share plant equipment and resources in a dynamic manner makes production planning the most vital component to this endeavour. An optimal production plan requires intelligent decision making in an integrated manner for campaign scheduling (involving numbers of batches, times, and sequences), cleaning and set-up, plant maintenance, testing and production of NCEs (new chemical entities), resource allocation, manpower handling, and inventory management. It requires a planner to evaluate a huge number of operational configurations, process constraints, statistical combinations, and business scenarios through iterative consultations with multiple departments.

The importance and need for intelligent and sophisticated planning tools or models aptly justifies the increasing interest of researchers in addressing the problem of operational planning. Mauderli and Rippin [87, 134] were among the first to address the problem of planning and scheduling in batch plants. They pointed out the importance of campaign mode operational strategy in planning and scheduling of batch processes. In their solution procedure, they generate alternative routes for producing each product followed by identifying a set of dominant campaigns. Finally, the problem of assigning timings to the dominant campaigns was solved using a linear program (LP). Later, Lazaros et al. [135] considered the effect of limited utilities in their approach using an approach of identifying dominant campaigns and production lines for planning, similar to Mauderli and Rippin [87, 134]. Birewar and Grossmann [55] presented an aggregated LP model for multi-period planning of multiproduct batch plants and considered no-intermediate storage (NIS) and unlimited intermediate storage (UIS) configurations for all materials. Wellons and Reklaitis [89] developed a MILP planning model for multipurpose batch plants. They presented a decomposition strategy to identify dominant campaigns and production lines that shown to be better than that from Mauderli and Rippin [87]. A simultaneous campaign formation and planning problem considering limited utilities and NIS/UIS configurations was solved by Shah and Pantelides [91] using a MILP formulation. This problem was then generalized for any general processing network with limited utilities and solved using a cyclic scheduling algorithm by Shah et al. [35]. Papageorgiou and Pantelides [92] used this cyclic scheduling algorithm to present a three-step approach for planning multipurpose batch plants. In the subsequent work [95], they extended their model considering mixed storage policies, shared intermediates, and limited resources. Bassett et al. [31] presented a number of decomposition approaches based on rolling horizon heuristics to solve large scale scheduling problems. McDonald and Karimi [13] and Karimi and McDonald [60] developed MILP models for planning and scheduling of semi-continuous processes. They modelled time in terms of slots of variable lengths and include minimum run lengths, various storage configurations, inability to fulfil demands, and material transports. Gupta and Maranas [136] used the model of McDonald and Karimi [13] and presented a hierarchical solution approach based on Lagrangean relaxation techniques. Oh and Karimi [137, 138] presented a production planning model for a single stage processor. Grunow et al. [120] presented a hierarchical decomposition approach for campaign planning and resource scheduling using a MILP formulation. Kallrath [139], Shah [140, 141], and Varma et al. [142] have presented comprehensive reviews highlighting the existing approaches, emerging research challenges, and the need for further work. Suryadi and Papageorgiou [143] considered a production planning problem along with the design of multipurpose batch plants and incorporated maintenance planning and crew allocation constraints. Sundaramoorthy and Karimi [144] studied the effect of new product introductions in the medium-term planning for a pharmaceutical production facility. They further assessed the feasibility or profitability of introducing new intermediates/products and outsourcing of the existing intermediates. Stefansson et al. [126] presented a 3-level hierarchical framework for an integrated problem of planning and scheduling in the context of secondary pharmaceutical plants. They modelled constraints to avoid the usage violation of limited resources. Verderame and Floudas [145] presented a discrete-time planning model for a multi-site production and distribution network. Recently, Corsano et al. [146] presented a MINLP model for the integrated problem of design and planning for multiproduct batch plants.

2.3.3 Global Integrated Planning

The long-time horizons involved in the planning problems make the integration of several factors necessary, as this would lead to a better operational planning. There has been a considerable research progress in this direction. Researchers have considered the planning problem with several factors such as NPIs, capacity planning, out-sourcing and sourcing of key intermediate or raw materials, revenue flows, contract selection, drug pipeline planning, etc. However, the work till now can be segregated as follows based on the scope of the problem that has been addressed.

Perhaps, one of the most important tasks in the pharmaceutical supply chain is the planning/scheduling and resource allocation in the R&D pipelines. The R&D pipeline involves various strategic and tactical decisions on the portfolio selection, resource allocation to various projects undertaken, and the subsequent study of the impact of its inclusion into the production. This deals mostly with the product development stage in the pharmaceutical supply chain. The selection of the portfolios is subject to various stochastic factors, which include market risks, clinical trial uncertainties, etc. The activity duration of a project in the pipeline depends on the quantity of resource allocated [147]. Schmidt and Grossmann [148] developed MILP models for scheduling tasks involved in the testing phase of agrochemicals and pharmaceutical products. While, they assumed unlimited availability of the resources, Jain and Grossmann [149] extended these models to consider resource constraints and utilize the option of outsourcing. In their model each task is attributed by a duration, cost, precedence, resource requirements, and success probabilities. Also, the income for a product was expressed as the function of time-to-market. However, their model enforces resource constraints even if a certain task fails and is now no more occurring. This restricts the scope of the model for no more tasks are further allowed to begin. Blau et al. [150] employed a heuristic rule based approach using simulation, which coupled risk management at the development stage. Maravelias and Grossmann [151] developed a multi-period capacity planning problem integrated with the scheduling decisions of the testing tasks. To reduce the problem size they proposed a heuristic based Lagrangean decomposition method. Subramanian et al. [152, 153] developed a simulation-optimization framework for R&D pipeline management, which includes uncertainty of clinical trials and resource allocation strategies. Blau et al. [154] addressed a similar problem to optimize the drug portfolio selection and decision prioritization, including the aspect of interdependent outcomes. Verma et al.[142, 155] extended this work by studying deeply the interaction between the resource allocations and its effect on the project durations. Colvin and Maravelias [156] developed a multi-stage stochastic programming framework for including testing trial planning in the new product development. Recently, they extended their framework [157] to include capacity expansions and outsourcing decisions. However, their model does not allow the testing tasks of the same project parallel, for which the current formulation needs to be extended to make it as a general resource constrained problem.

Global integrated enterprise-wide planning has attracted some interest from the academic community with some work on pharmaceutical industry. McDonald and Reklaitis [158] highlighted the importance of considering financial aspects such as taxes, duties and transfer pricing in supply chain optimization models. Grossmann [159, 160] and Varma et al. [142] reviewed in detail the current research trends in enterprise-wide optimization and highlighted current challenges and emerging future challenges. They stressed the need for developing novel computational models and algorithms to solve real-world problems and strengthen the economic performance and competitiveness of the process industries. Shah [141], Barbosa-Povoa [161], and Papageorgiou [162] reviewed existing models and key issues in pharmaceutical supply chains.

Cohen and Lee [163] presented an enterprise-wide optimization model for a company operating in batch mode, and determined costs for multiple operational scenarios. Timpe and Kallrath [164] developed a MILP model for optimizing a multi-site network with production, distribution, and marketing constraints. However, the model was difficult to solve for large problems. Thus, a need exists for developing efficient solution strategies for large problems. Papageorgiou et al. [165] and Gatica et al. [166] developed models for capacity expansion, production, and distribution under uncertainties for pharmaceutical enterprises. Papageorgiou et al. [165] also considered tax differentials and transfer prices using a scenario-based approach in their planning problem. Later, Levis and Papageorgiou [167] developed a similar capacity expansion model that considers product development tasks, product success probabilities, and demand uncertainties. Oh and Karimi [168, 169] studied the impact of considering regulatory affairs and duty drawbacks at the planning stage for facility selection, investment profiles, sourcing decisions, etc. Sundaramoorthy et al. [170] developed a simple LP model as a decision support tool for medium-term integrated planning decisions in the pharmaceutical and the specialty chemical industry. Their model integrates different layers in the supply chain of a global company to enable sourcing, production, transfer, and distribution decisions. The model considers production details along with constant transfer prices, and accounts for various costs such as holding, transportation, and backlog. Ryu and Pistikopoulos [171] considered an enterprise-wide production and distribution model under different operational policies. Naraharisetti et al. [172] developed a model for supply chain redesign, asset management, and capital budgeting. They included several factors such as disinvestments, technology upgrades, supply contracts, capital raising loans and bonds, transportation costs, and shutdowns.

Bok et al. [173] presented a MILP model for multi-period planning of continuous processes. They improved on the short-term supply chain optimization problem of Norton and Grossmann [174] by considering additional constraints such as inventories, changeovers, and demand violations. The authors then proposed a bi-level decomposition-based solution strategy to enhance computational efficiency. Jayaraman and Pirkul [175] developed an integrated model for locating production and distribution facilities. They considered a supply chain consisting of several nodes from raw material vendors to customers. To solve the MILP model, authors proposed a heuristic solution strategy using a Lagrangean relaxation methodology. Park [176] presented MILP models for both integrated and decoupled multi-site, multi-product production–distribution problem. He then proposed a heuristic solution strategy to solve large problems. Amaro and Barbosa-Povoa [118] developed a strategy for the integration of planning and scheduling models in pharmaceutical supply chains. They developed two MILP models and solved them sequentially. Sousa et al. [177] developed two-level hierarchical models for supply chain redesign and production/distribution planning in an agricultural company. Eskigun et al. [178] developed a MILP model for a vehicle distribution logistics problem. They presented a Lagrangean heuristic to improve the solution time for solving an industrial case study. Later, Chen and Pinto [129] presented a number of decomposition techniques for a continuous flexible process network model similar to Bok et al. [173]. The techniques include Lagrangean decomposition, Lagrangean relaxation, and Surrogate relaxation, coupled with sub-gradient and modified sub-gradient optimization. You and Grossmann [179] presented a MINLP model for integrated inventory management, transport management, and network design in a multi-echelon supply chain considering uncertainty in customer demand. They proposed spatial decomposition algorithm based on Lagrangean relaxation and piecewise linear approximation to enhance computational performance. You et al. [180] used Lagrangean and bi-level decompositions to solve an integrated problem for capacity, production, and distribution planning in a multisite specialty chemical company. Sousa et al. [181] developed a multi-period planning model for multinational pharmaceutical companies taking both primary and secondary manufacturing plants into consideration. The authors then studied two decomposition schemes (structural and time-based) to reduce solution time.

2.4 Tools for planning and scheduling pharmaceutical plants

Planning and scheduling activities in the pharmaceutical industry are crucial. The aforementioned review of the existing literature emphasizes that a significant effort, from the academic community, has been made in two different directions to address this complex problem. In the first, the development of generic, comprehensive, and complex mathematical models have been the area of focus. Here, the prime objective has been to develop models that closely resemble the real-life operations and obtain good if not the best solutions. However, for large scale examples, such models usually are either very difficult or are computationally intractable. Consequently, the second direction of research has mainly focused on the development of a variety of algorithms and solution strategies for large-scale or real-life problems. Here, the focus has been primarily on solving large scale models and developing strategies or developing simpler mathematical models to obtain good solutions quickly. However, industrial applicability, ease-of-use, and the specificity to pharmaceutical plants collectively have received a rather little attention in both academic literature and industrial practice. Currently, a few tools are available commercially for the aforementioned problem of planning and scheduling of process industries in general such as Oracle ERP, SAP ERP and other problem specific software packages. Such commercial solutions although are very popular in general chemical process industries, they have not found a wide application in the pharmaceutical companies. This is mainly because of the following reasons. The commercial software packages are usually very generic in nature and so, they fall short in addressing several specific constraints particular to the pharmaceutical operations. Also, such available tools are generally modular and have a wider application than just planning and scheduling. However, the specific modules related to planning and scheduling in such packages are mostly transactional in nature and not very efficient in solving complex pharmaceutical problems. They offer very limited freedom to the user in changing the problem configuration to assist a comprehensive and scenario based study. Thus, there have been some significant contributions from a variety of industries in developing better and efficient tools.

Karmarkar [182] and Bayer et al. [183] developed and presented a revolutionary method based on interior point algorithm for optimizing resource allocations and operational parameters in general resource intensive plants. This method is useful in solving large problems quickly, which otherwise were difficult to solve. Dembo [184] presented a tool to systematically optimize resource allocations in the presence of parametric uncertainties. He developed a scenario-based optimization approach for modelling and evaluation of optimal solution under different possible occurrences of an uncertain event. Kurihara et al. [185] presented a mathematical programming based job scheduling model for batch operations. They used a discretized time approach for short-term scheduling. To overcome the limitations of time-discretization approach, authors implement an additional algorithm to evaluate alternative schedules iteratively for improvement. Diezel and Finstad [186] presented a detailed scheduling model with constraints on the start/end and duration along with the precedence relationships among various activities. They considered limited availability of resources required and allocated them in the order of priority to each activity. Dietrich and Wittrock [187] developed an integrated LP based method for material requirement planning, resource allocation, and production planning for a resource intensive plant. They used maximization of the plant's profit as an objective function and considered a number of constraints such as product demands, raw materials inventory, and resource availability. Trautmann and Schwindt [188] proposed a multi-level hierarchical approach for solving a resource constrained short-term scheduling problem for multipurpose batch plants. Their system adopts a mathematical model based on task precedence constraints for scheduling batch plants. This model considers resources such as human, equipment, and materials with constraints on material shelf-lives, order release times, and availability of operators. Although the algorithm does not guarantee an optimal solution, it may be useful in finding a good solution. Strain et al. [189] developed an integrated operational design and scheduling system for a batch plant. Their system consists of different modules such as data, design, scheduling, and quality. These integrated modules allow decisions pertinent to scheduling, material purchasing, and inventory monitoring. Although this system is appropriate for a general batch plant but it does not address the specific constraints of pharmaceutical plants such as scenarios evaluation, multi-step and multi-stage manufacturing, and campaign mode operations. Also, it does not study the effect of resource allocation on the process performance. Kataria et al. [190] presented an integrated inventory monitoring and compliance and tax reporting system. For this, they develop a web-based system that essentially manages an entire pharmaceutical supply chain including product history for all materials from raw to final products. Goodall et al. [191] presented a system for effective management of workload among a multiple pharmacy network system that are connected through a common information system such as internet. The system is specifically designed for a drug distribution network consisting of entities such as dealers, retailers, and consumers including a number of specific constraints such as quality of equipment, manpower, etc. Popp [192] presented a quality check, risk assessment, and production monitoring system for pharmaceutical companies. His tool finds its use at several stage of drug production from the phase of clinical trials to its distribution to consumer markets. Couronne et al. [193] presented a mathematical model based system for production planning of batch plants. Their model minimizes inventory by simultaneously calculating safety stock quantities with production planning.

2.5 Summary of gaps and challenges

Evidently, the aforementioned literature review shows that a significant progress has been made in the area of planning and scheduling batch plants in general and pharmaceutical plants in particular. Again, based on this review, we identify some research gaps and conclude the following challenges and opportunities.

(1) The problem of short-term batch scheduling is well studied in the literature. Also, a number of approaches such as events-, slots- (synchronous and

asynchronous), precedence-, unit-specific events-, single-, multi-, or no-grid construction of time have been developed and used. However, recent developments [80, 82, 86] have shown the limitations of multi-grid approaches (asynchronous-slots or unit-specific events). The single-grid (synchronous-slots and global events) and precedence based (general and immediate) formulations are difficult to solve because of the large number of variables and constraints. Thus, construction of an efficient and general model for short-term scheduling still remains an open problem.

(2) Majority of work in the literature is focused on formulating a simplified model by considering only one or more of a number of real batch plant characteristics. For instance, a majority of models consider materials and equipment as only resources in a complex batch plant. Such models simply the real problems by making critical assumptions such as simultaneous and instantaneous material transfers, sequence-independent transition or setup times, no discrete resources (e.g. human), unlimited waste storage and treatment capacity, etc. This is essentially due to the highly complex nature of batch plants. However, these assumptions hinder the application of such models to practical problems, as they do not assist in generating practically feasible schedules. Clearly, further work is required to develop more comprehensive and efficient approaches that consider a number of practical features in scheduling batch plants.

(3) Resource allocation is a critical element in production planning. Usually, operations can be expedited or impeded by controlling the amount of resources allocated to them. Thus, a given production plan may either be an over-estimate or an under-estimate to the actual scenario if it does not consider resource scheduling constraints. Only a few of the existing works consider resource

scheduling constraints along with the production planning. Also, an integrated approach for resource allocation and campaign planning under various decisions such as responsive scheduling for including clinical trials, optimal manpower allocation, maintenance planning, etc. does not exist in the literature. In practice, planners prepare and evaluate multiple production plans based on different scenarios. Thus, the flexibility of a model to allow the generation of multiple production plans based on different resource allocation profiles and market conditions is important.

(4) A holistic and integrated decision making at the enterprise level considering the nuances of individual entities and functions along with their complex interactions is extremely difficult and critical for the economic sustainability of a pharmaceutical company. While some works in the literature have addressed the integrated problem of procurement, production, and distribution in the context of multinational pharmaceutical companies, the focus has been mainly on developing better solution strategies that improve computation time for large scale problems. However, there is a need for simpler models that are easy to implement, quick to solve, but do not compromise problem realism or features.

2.6 Research Focus

Based on the above challenges, this research project focuses on the following aspects.

1. Some recent scheduling models are analysed. Their limitations and suggestions to address such limitations are discussed. It is shown that by not addressing these limitations, such models in some cases may lead to non-optimal

solutions. Three examples involving shared and limited storage for short-term scheduling of batch plants are presented to demonstrate these findings.

2. A novel approach to scheduling multipurpose batch plants using unit-slots instead of process-slots to manage shared resources such as material storage is presented. Here, two slightly different but compact and simple models are developed. This multi-grid approach rationalizes, generalizes, and improves the current multi-grid approaches for scheduling with shared resources. Also, the models allow non-simultaneous transfers of materials into and out of a batch, which is shown to give better schedules than those from existing models in some cases. Furthermore, the presented approach requires fewer slots (event-points) on some examples than those required by the unit-specific event-based models.
3. Extends and generalizes two different models one is the multi-grid scheduling approach based on unit-slots and the other is the single-grid approach based on process-slots to consider rigorous resource constraints. Here, a number of real-life scheduling considerations such as sequence-dependent set-ups, effects of resources (other than material and equipment) on scheduling, non-simultaneous material transfers, non-zero transfer times, and multiple storage configurations are incorporated. In addition, different variations for the presented models that appropriately suite their application to a given problem are discussed.
4. The concept of resource availabilities affecting production scheduling is further generalized and studied with a strategic perspective. A framework is developed to study the effect of resource allocation on the process performance. Also, a few key aspects of the industrial planning activity such as interactions among the planner and other stakeholders, campaign mode operations, and safety stock

policy are considered. A simple mathematical model for integrated resource allocation and campaign planning is presented. The model enables decision support pertaining to campaign scheduling, sequence-dependent changeovers, key resource allocations, scheduled maintenance, inventory profiles with safety stock limitations, and new product introductions.

5. The integrated problem of resource allocation and campaign planning is generalized from a single plant to the entire production supply chain of a multinational pharmaceutical enterprise. A simple yet powerful model for multi-period enterprise-wide planning is presented. Here, the entire enterprise is represented in a seamless fashion with a granularity of individual task campaigns on each production line. The model considers an integrated problem of procurement, production, and distribution and incorporates several practical features of industrial planning such as effects of international tax differentials, inventory holding costs, material shelf-lives, waste treatment / disposal, and other real-life factors on the after-tax profit of a company.
6. A decision-support tool for integrated production planning and resource allocation in pharmaceutical plants is presented. The tool is developed in association with a Singapore-based plant of a multinational pharmaceutical company. PlanPerfect is motivated from the existing complex problem of production planning at the associated plant. It is specifically designed and customized to address the needs and constraints of planners in any pharmaceutical plant.

Apart from the aforementioned research issues in focus, one of the main problems pertaining in the process industries is the uncertainty and disruptions of operations. Although we do not address uncertainty explicitly in this research project, we discuss strategies based on reactive scheduling to deal with disruptions in each of our work. A lot of work has, however, been done in the area of addressing process uncertainties in process plants, which can be directly applied to the problems we consider in this work. We request the readers to refer appropriate literature to know more about such approaches. In an earlier section, we have presented a detailed literature review of the approaches existing for reactive scheduling methodology. Also, we discuss along with the specific problems presented later in this thesis, the application of such reactive scheduling approaches with respect to our defined problems.

3 AN ANALYSIS OF SOME MULTI-GRID SHORT-TERM BATCH SCHEDULING MODELS[1, 2]

3.1 Introduction

Optimal scheduling of operations in a batch plant is known to be a pivotal problem and so, has received a significant attention [12, 14, 41] in the literature. In the last three decades, several techniques and models have been developed to solve the problem of short-term batch scheduling. This include a variety of discrete-time [34, 194, 195] and continuous-time formulations based on the way such models handle time. Figure 3.1 presents the three types of continuous-time scheduling models used in the literature.

As discussed in the previous chapter, the slot-based models [46, 64, 65, 68, 71] model time by means of ordered slots of non-uniform and unknown lengths to which batches, tasks, or activities are assigned. The literature has used two slot types. If a single common or shared set of slots is used for all units in the process, then such slots have been called synchronous [63] or process slots [64]. If an independent or separate set of slots is used for each unit in the process, then such slots have been called asynchronous [63] or unit slots [64]. The sequence-based models [42, 43, 45, 47] use direct

[1] Li, J., Susarla, N., Karimi, I. A., Shaik, M., & Floudas, C. (2010). An Analysis of Some Unit-Specific Event-Based Models for the Short-Term Scheduling of Noncontinuous Processes. Ind. Eng. Chem. Res, 49, 633-647.

[2] Susarla, N., Li, J., & Karimi, I. A. (2008). A novel continuous-time formulation for short-term scheduling of batch processes. Presented in AIChE Annual Meeting, Philadelphia, PA, USA.

(immediate) or indirect (general) sequencing (precedence) of task-pairs on units to define a schedule. They do not model time explicitly in terms of slots or event points. While this eliminates the need to postulate the numbers of slots or event points a priori, they must postulate the number of tasks a priori. The global event-based models [75, 76, 196] use one single set of event points and times for all units in a process. These are analogous to the models using process slots. The unit-specific event-based models [74, 78, 80-82, 85, 114, 197-205] "introduce an original concept of event points, which are a sequence representing the beginning of a task or utilization of the unit. The locations of event points are different for different units, allowing different tasks to start at different moments in different units for the same event point" [12]. While the two approaches using unit-specific events or unit-slots are analogous, the noteworthy contribution of unit-specific event-based models is in handling shared resources without using any additional binary variables.

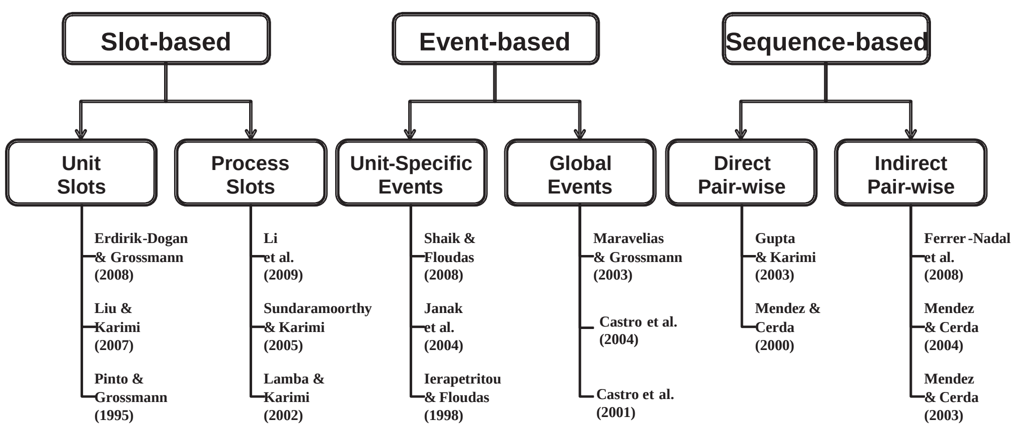

Figure 3.1 A classification of continuous-time scheduling models

The partially independent and asynchronous locations of event-points on different units and one event point per task enable some unit-specific event-based models [78, 85, 205] to use fewer event-points (thus binary variables) than the process slots or global event-based models. The models using process slots or global event points require equivalent numbers of slots and event points (thus binary variables) respectively. As model solution times generally grow exponentially with the number of slots or event points, some unit-specific event-based models have proved much faster [78, 85, 205] than other models. Recently, Shaik and Floudas [82] demonstrated that with tasks spreading over multiple events also, their unit-specific event-based model performs better than the process slots or global-event based models for several examples.

Since their introduction by Ierapetritou and Floudas [74, 199] and Ierapetritou et al. [200], the unit-specific event-based models or their variants have been used to solve a variety of problems in process operations. A key feature of these models is that the resource balance is done over event points with no unique time values. When the time instances linked to a given event point vary from unit to unit, such a balance in the domain of event points rather than time may lead to discrepancy in resource balance as shown in the literature [78, 80, 82, 205]. As discussed in Floudas and co-workers [78, 80, 82] both the original model of Ierapetritou and Floudas [74] and their improved model as given by Shaik et al. [85] are not applicable for short-term scheduling of batch plants with finite intermediate storage and may yield real-time storage violations. To prevent this, either these models [78, 82] employ additional task sequencing constraints to align the event timings across different units or consider storage as a separate task as done by Lin and Floudas [204]. While these constraints have correctly solved several literature problems, Janak et al. [81] introduced an additional construct of storage tasks to guarantee that they work correctly irrespective of scenario, example, or data. This was a significant departure from other models [74, 78, 85, 114, 199, 200, 204] in that the additional storage tasks map the various storage and processing tasks on to a single time line and a task may span multiple event-points. Indeed, Janak and

Floudas [80] and Shaik and Floudas [82] demonstrated that not allowing tasks to span over multiple event-points might yield suboptimal solutions in some cases. To reduce the complexity and improve the efficiency of the model of Janak et al. [81], Shaik and Floudas [82] proposed a novel unified model that also allows tasks to occur over multiple event points. Their model requires an extra set of iterations that control the number of event points that a task is allowed to span. Shaik and Floudas [82] established that both the original model of Ierapetritou and Floudas [74] and their improved model as given by Shaik et al. [85], the model of Lin and Floudas [204] with storage tasks; and the RTN-based model of Shaik and Floudas [78] may give suboptimal solutions in some cases since they do not allow tasks to occur over multiple events. For short-term scheduling of semi-continuous plants, Shaik and Floudas [205] presented an improved model compared to Ierapetritou and Floudas [199] that performed better than the other models considered in their study.

In this chapter, we present and analyze two examples involving batch processes to study the performance of some recent variants of unit-specific event-based models. These examples involve batch plants with finite storage for some intermediates. Both the original model of Ierapetritou and Floudas [74] and their improved model as given by Shaik et al. [85] are not applicable for finite storage cases, unless storage is considered as a separate task as reported in literature [78, 81, 82]. However, these models [74, 85] have been shown [82] to yield suboptimal solutions for some unlimited storage cases also due to tasks occurring over single events. So, in this study, the RTN-based model of Shaik and Floudas [78] is used to solve the first two examples for which it gives either trivial or suboptimal solutions depending on the example and data. This behavior is not unexpected since finite storage here acts as a shared resource, and it has been shown in the literature [80, 82] that tasks should be allowed to span over multiple events to avoid discrepancies when there are shared resources. More importantly, we show that the recent model of Shaik and Floudas [82] indeed addresses these shortcomings and successfully solves these two examples on multipurpose batch plants.

3.2 Models and Implementations

For this study, we used CPLEX 10.0.1/GAMS22.2 on a Dell precision PWS690 workstation with Intel® XeonR 3 GHz CPU, 16 GB RAM, and Windows XP Professional x64 Edition operating system. We implemented the model of Shaik and Floudas [78] on this computing platform.

To validate our implementations, we solved the following literature examples and confirmed that their solutions match those given by the unit-specific event-based models in the literature.

1. Example 1 of Sundaramoorthy and Karimi [70]
2. Scheduling example from Kondili et al. [34]

The model statistics (optimal objective value, binary/continuous variables, constraints, etc.) in our implementations match those reported in the literature by various unit-specific event-based models. This gives further support to the validity and accuracy of our implementations.

We now present the five test examples and discuss them in detail one by one. For each example, we report the solution obtained from our implementation of an appropriate unit-specific event-based model, and another from a new model by Susarla et al. [206] that uses unit slots. The latter can also be obtained manually or by using a scheduling model that uses global event points [76] or process slots [70]. The two examples on batch plants show how the optimal solutions can be obtained using the unified model of Shaik and Floudas [82].

3.3 Example 1

The first example (Figure 3.2) is a modification of the motivating example of Maravelias and Grossmann [76]. The modification is that the storage capacities of tanks S2 and S3 (holding hA and IB respectively) are 6 kg and 4 kg instead of unlimited. Table 3.1 gives all the data. The scheduling objective is maximum profit over a horizon of 6 h, which is equivalent to maximum production for this example. We consider two scenarios. Scenario A assumes fixed batch sizes of 10 kg for H (Heater), 4 kg for R1 (Reactor-1), 2 kg for R2 (Reactor-2), and 10 kg for C (Separator). Scenario B allows the flexibility of using any batch size lower than the one assumed in scenario A.

Table 3.1 Data for Example 1

		batch size (kg)		
task/unit	duration (h)	scenario A (fixed)	scenario B (variable, max)	max capacity (kg)
task 1/heater	1	10	10	-
task 2/reacotr-1	3	4	4	-
task 3/reactor-2	1	2	2	-
task 4/separator	2	10	10	-
intermediate storage tank-S2	-	-	-	6
intermediate storage tank-S3	-	-	-	4
scheduling horizon	6	-	-	-

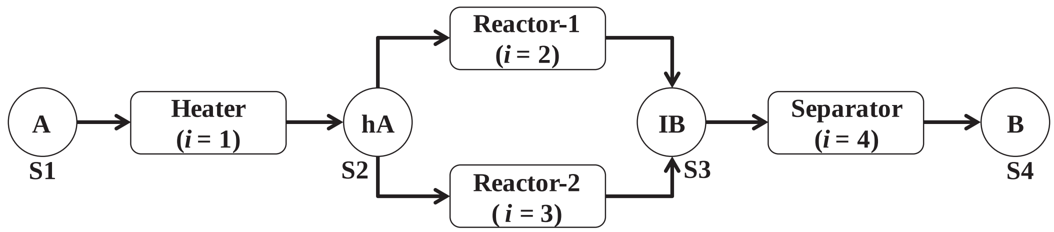

Figure 3.2 State-task network for Example 1

For scenario A, the model of Shaik and Floudas [78] yields a trivial schedule with zero production. This is irrespective of the number of event points. While this model is saying that no production is possible in 6 h, a feasible schedule (Figure 3.3) with a production of 10 kg in fact exists. In this schedule, H produces a batch of 10 kg at 1 h. From this, 4 kg go to S2, 4 kg go (directly) to R1, and 2 kg go to R2. Both R1 and R2 begin a batch at 1 h. At 3 h, R2 finishes two batches of 2 kg each, so S3 holds 4 kg of IB. At 4 h, R1 finishes its first batch of 4 kg and R2 finishes its third batch of 2 kg. This 6 kg along with the 4 kg from S3 enable a batch of 10 kg for the separator, which ends at 6 h. Figure 3.4 presents the inventory profiles of s2 and s3 for the schedule in Figure 3.3. The holdups of s2 and s3 always respect their storage capacities of 6 kg and 4 kg respectively at all times during the scheduling horizon. Thus, the schedule in Figure 3.3 is feasible and produces 10 kg of product.

To analyse why some unit-specific event-based models fail to generate the above schedule, we recall three critical features or constraints.

(1) Each task is associated with only one event point. In other words, a task that starts at an event point also ends at the same event point. Of course, its actual start and end times will be different.

(2) The resource balance is:

$$ST(s,n) = ST(s,n-1) + \sum_{i\in\rho_{si}>0} \rho(s,i) \sum_{j\in suit_{ij}} b(i,j,n-1) + \sum_{i\in\rho_{si}<0} \rho(s,i) \sum_{j\in suit_{ij}} b(i,j,n) \quad \forall\ s,n \tag{3.1}$$

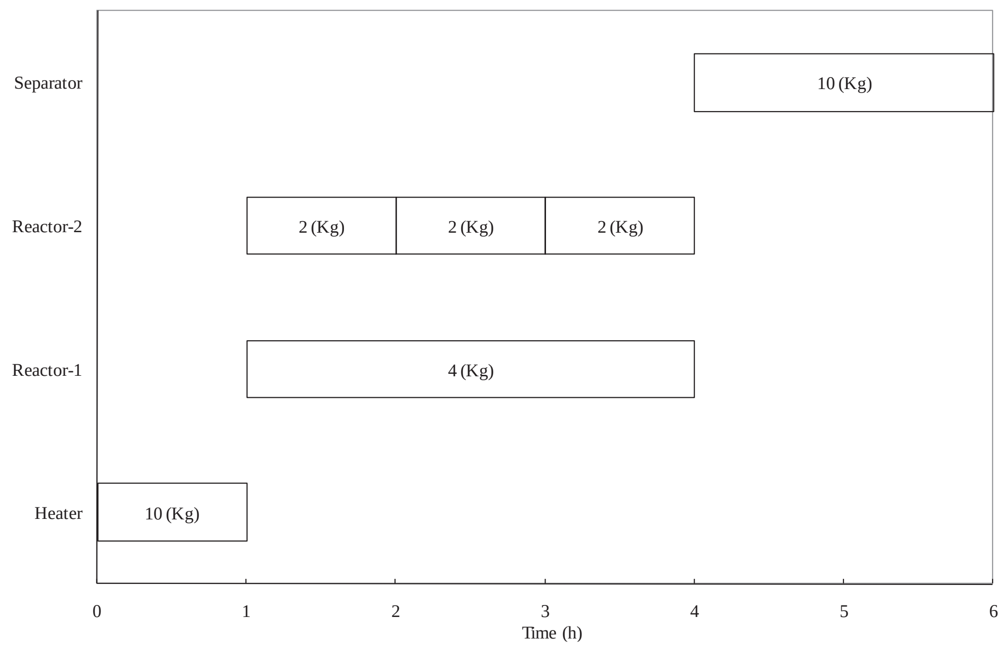

Figure 3.3 Optimal schedule for Example 1 obtained from Susarla et al. [206]

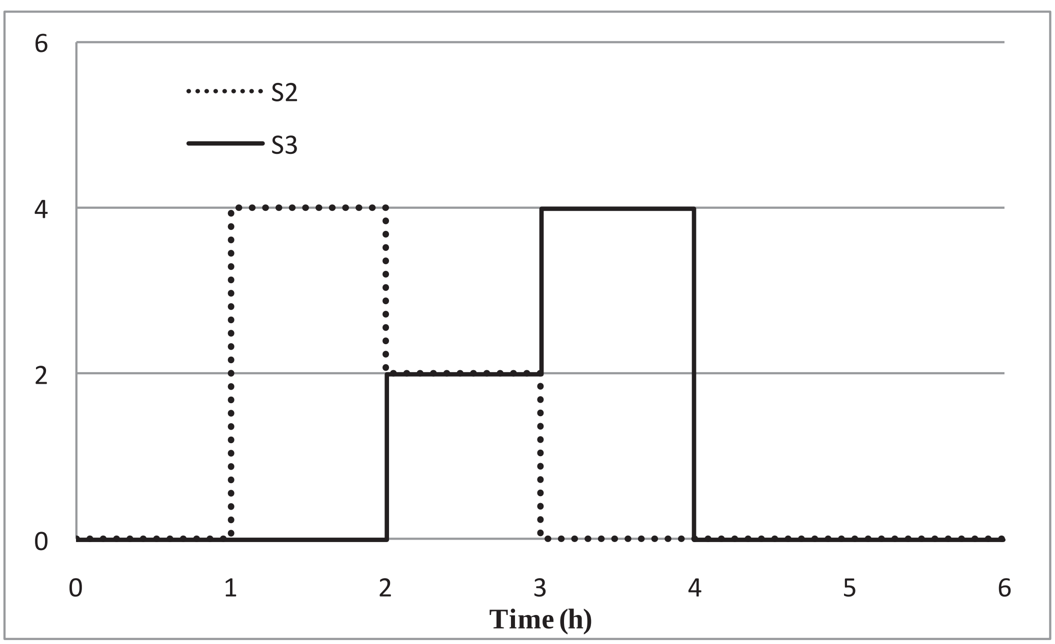

Figure 3.4 Inventory profiles of s2 and s3 for the schedule in Figure 3.3

$$E(r,n) = E(r,n-1) + \sum_{i \in I_r} (\mu_{ri}^{p} w(i,n-1) + \rho_{ri}^{p} b(i,n-1)) + \sum_{i \in I_r} (\mu_{ri}^{c} w(i,n) + \rho_{ri}^{c} b(i,n))$$

$$\forall\ r \in R,\ n \in N,\ n > 1 \qquad (3.2)$$

Note that eq. 3.1 is from the state-task network model of Shaik et al. [85] and eq. 3.2 is from the resource-task network model of Shaik and Floudas [78]. These two mass balance equations can be represented by the following equation:

Level (n) = Level (n–1) + Resource Generation (n–1) – Consumption (n) (3.3)

where, n denotes an event point.

(3) Level (n) of each resource must satisfy the specified upper/lower limits on its capacity. For a shared resource such as the storage of a material, the material inventory at each event point must be within zero and the storage capacity.

With these three features in mind, let us now analyse the solution for this example. The heater begins a batch of 10 kg at n = 1 (or n1) and produces 10 kg of hA at n1. Since S2(n1) = 0, R1 and R2 cannot start a batch at n1. Since H produces 10 kg at n1 and S2(n2) cannot exceed 6 kg, R1 must begin a batch at n2, because the batch size on R2 is only 2 kg. The batch from R1 makes S3(n3) = 4 kg, which is the maximum capacity of S3. Since the separator needs 10 kg to begin a batch, it cannot begin a batch at n3. However, the fact that S3 is full at n3 means that R2 cannot begin a batch even at n2. For the same reason, R1 cannot also begin another batch at n3. Furthermore, such a batch would not finish by 6 h. Thus, there is no recourse. Since S3 is full, R1 cannot begin another batch and R2 cannot begin any batch. On the other hand, since S3 does not have 10 kg, the separator cannot begin a batch. Thus, irrespective of the number of event points, the model cannot yield any schedule other than the trivial schedule with zero production. This shows that the model successfully solves (Figure 3.5) the original motivating example of Maravelias & Grossmann [76], but may lead to suboptimal or trivial solutions, when the example data change.

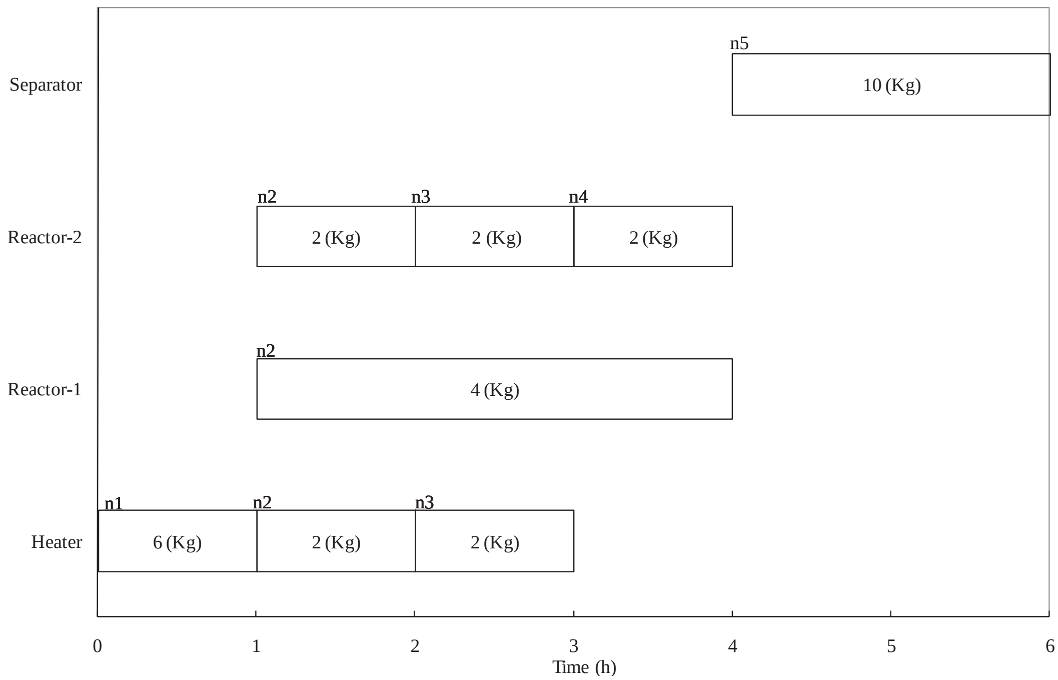

Figure 3.5 Optimal schedule for Motivating Example from Maravelias & Grossmann [76] from the model of Shaik & Floudas [78]

For scenario B with variable batch sizes, the model of Shaik and Floudas [78] yields the schedule in Figure 3.6 with a production of 8 kg. This is suboptimal, because a better schedule (Figure 3.3) producing 10 kg exists. In Figure 3.6, H begins a batch of 8 kg at n1 and produces 8 kg of hA at n1. S2(n1) = 0 prevents R1 and R2 from starting a batch at n1. If both R1 and R2 begin a batch at n2, then the separator must begin a batch of at least 2 kg at n3 to avoid the overflow in S3. This batch would end at 6 h, and we will have an inferior solution of at most 6 kg. Therefore, the optimizer begins a batch of 2 kg on R2 at n2 and S3(n3) = 2 kg. It would be useless for the separator to begin a batch of 2 kg at n3, so it must wait. Now, both R1 and R2 can begin a batch at n3. This would produce 6 kg of IB at n3 and with the 2 kg from S3, the separator can begin a batch of 8 kg at n4. Hence, the schedule in Figure 3.6 produces 8 kg with inventory profiles of s2 and s3 in Figure 3.7.

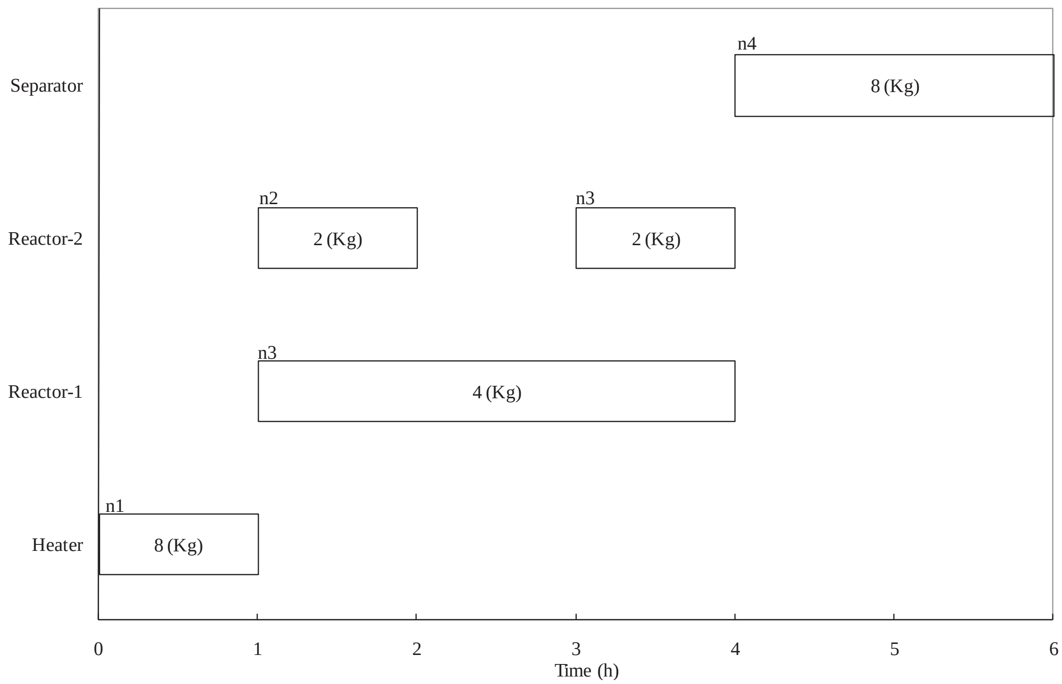

Figure 3.6 The schedule for Scenario B of Example 1 from Shaik & Floudas [78]

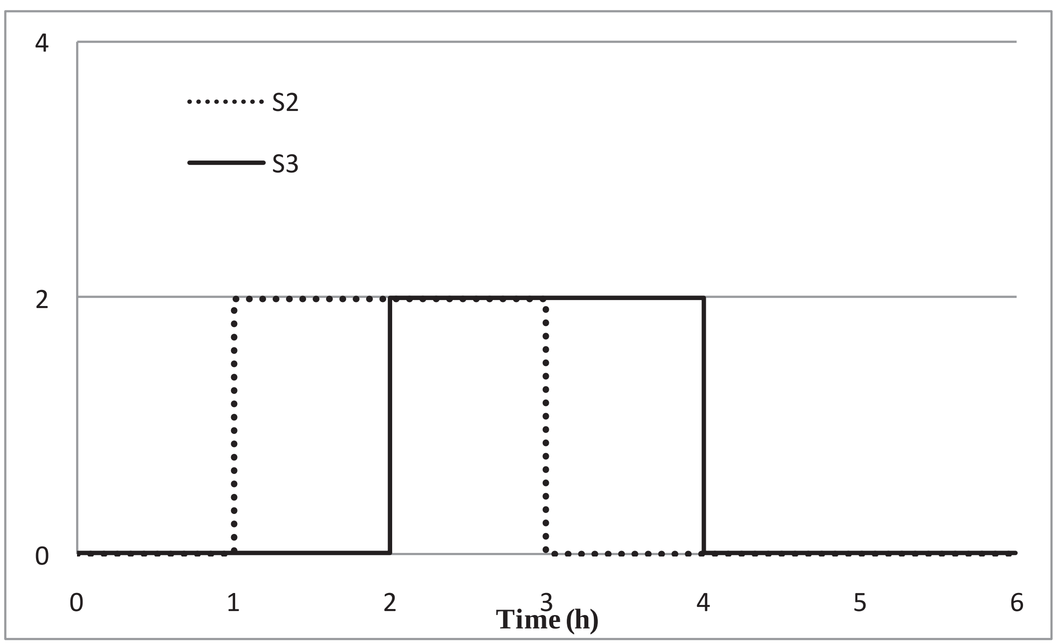

Figure 3.7 Inventory profiles of s2 and s3 for the schedule in Figure 3.6

Now, let us see why the Shaik and Floudas [78] model cannot generate a better schedule than Figure 3.3. Suppose that H begins a batch of 8+δ kg ($0 < \delta \leq 2$) at n1. The models have three options. The first is to start a batch on both R1 and R2 at n2. As seen earlier, this cannot give a solution better than 6 kg. The second is to begin a batch only on R2 at n2, but this makes S1 overflow at n2. Therefore, the last option is to begin a batch of at least 2+δ kg and at most 4 kg on R1 at n2. Having done this, R1 cannot begin another batch at n3, because that would go beyond the horizon of 6 h. Furthermore, R2 can begin a batch of at most 2 kg at n3 or any later event point. This would force the separator to begin a batch at the next event point. The size of such a batch cannot exceed 4+δ kg, and another batch on the separator would be impossible within the 6 h horizon. Therefore, if the heater processes 8+ δ kg, a solution with more than 6 kg of production is impossible. Thus, the schedule in Figure 3.6 is the best that this model can give.

Let us contrast our analysis for this example with that of Maravelias and Grossmann [76]. Recall that the data are different from those used by Maravelias and Grossmann [76]. Maravelias and Grossmann [76] assumed the first task (heating) to start at an arbitrary event point k. Then, they argued that the second task (reaction) must begin at the next event point (k +1) in both R1 & R2 simultaneously. They saw this as necessary to satisfy the mass balance equation. However, they did not allow the possibility that the second task could start at any subsequent event point (e.g. k+2, or later). Their assumption that the second task must begin at event point (k+1) resulted in a mass balance error later in the schedule. The example itself with the original data of Maravelias and Grossmann [76] shows no error. Janak et al. [81] confirmed this by solving the same example successfully using the Lin and Floudas [204] model by considering storage as a separate task, and we have also done the same (Figure 3.5) using the model of Shaik and Floudas [78]. In contrast, by modifying the example data, we have shown that some unit-specific event-based models may fail to give the optimal solution without any obvious mass balance error due to tasks occurring over single events.

3.4 Example 2

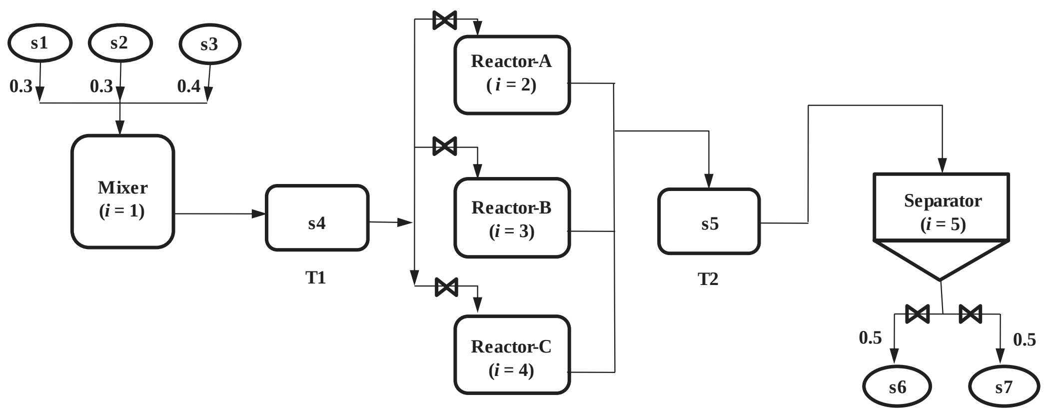

Figure 3.8 State-task network for Example 2

This example (Figure 3.8) involves five tasks, five processing units (Mixer, Reactor-A, Reactor-B, Reactor-C, Separator), two buffer tanks (T1 and T2), and seven materials (s1-s7). T1 and T2 store s4 and s5, with a capacity of 60 mass units (mu) each. The scheduling horizon is 9 h. Again, we consider two scenarios, scenario A with fixed batch sizes, and B with variable batch sizes. Table 2 gives all the data for this example.

Table 3.2 Data for Example 2

		batch size (mu)		
task/unit	duration (h)	scenario A (fixed)	scenario B (variable, max)	max capacity (mu)
task 1/mixer	1.5	150	150	-
task 2/reactor-A	4.5	60	60	-
task 3/reactor-B	1.5	30	30	-
task 4/reactor-C	1.5	30	30	-
task 5/separator	3	150	150	-
T1	-	-	-	60
T2	-	-	-	60
scheduling horizon	9	-	-	-

For scenario A, the model of Shaik and Floudas [78] yields the trivial solution with zero production irrespective of the number of event points. In contrast, the feasible schedule in Figure 3.9 produces 150 mu. Figure 3.10 gives the inventory profiles of s4 and s5 for this schedule. An explanation for this behaviour is similar to that given for Example 1. The capacities of T1 and T2 prevent any production in this model.

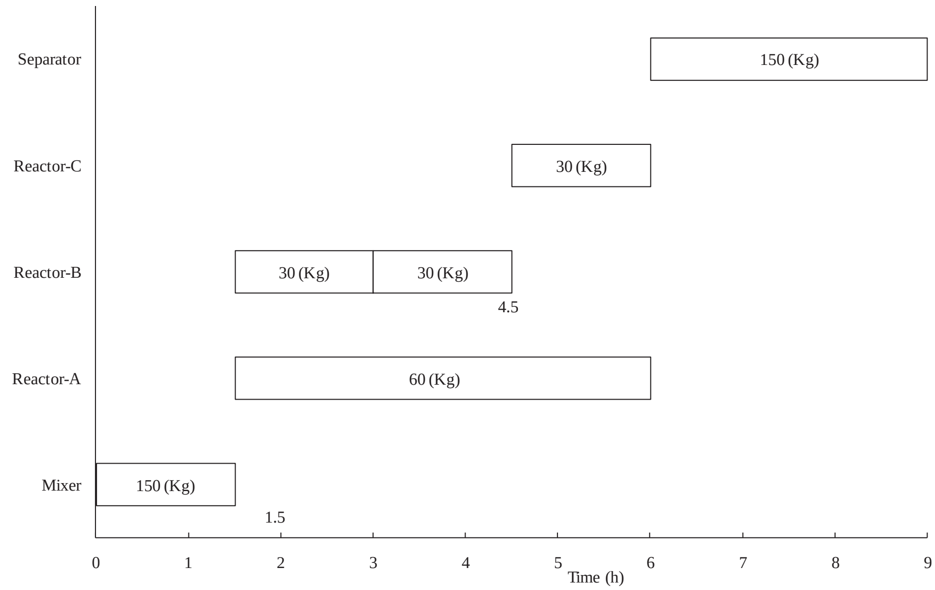

Figure 3.9 A feasible schedule for Scenario A of Example 2 obtained from Susarla et al. [206]

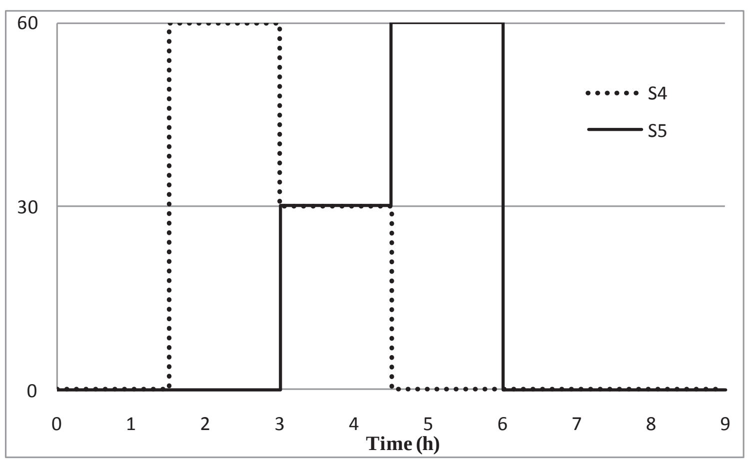

Figure 3.10 Inventory profiles of s4 and s5 for the schedule in Figure 3.9

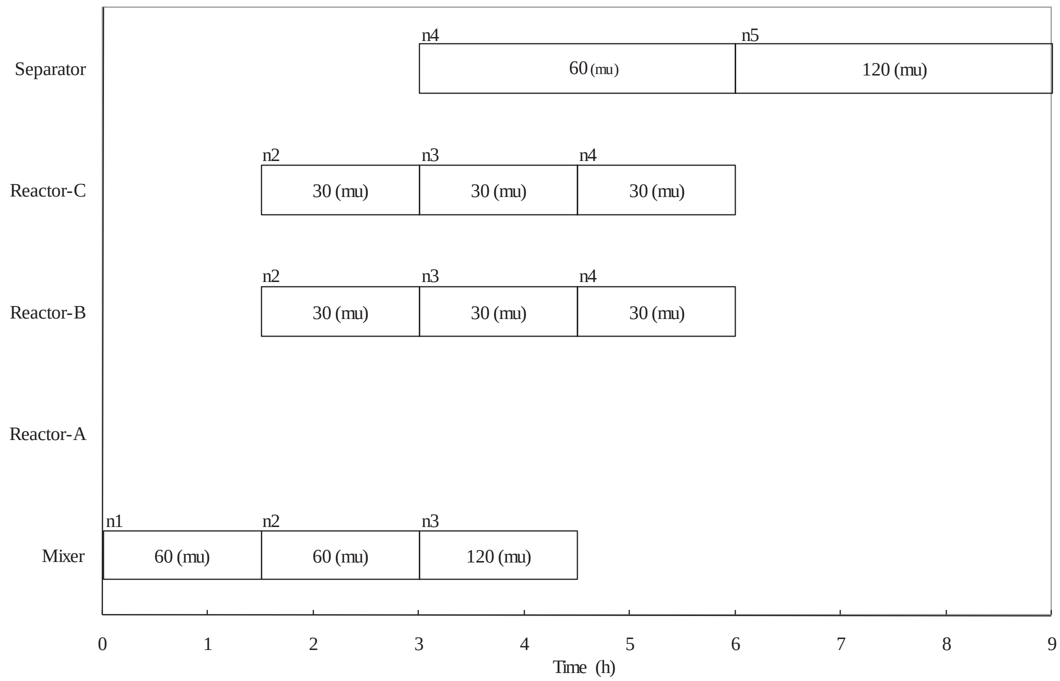

Figure 3.11 The schedule for Scenario B of Example 2 from Shaik & Floudas [78]

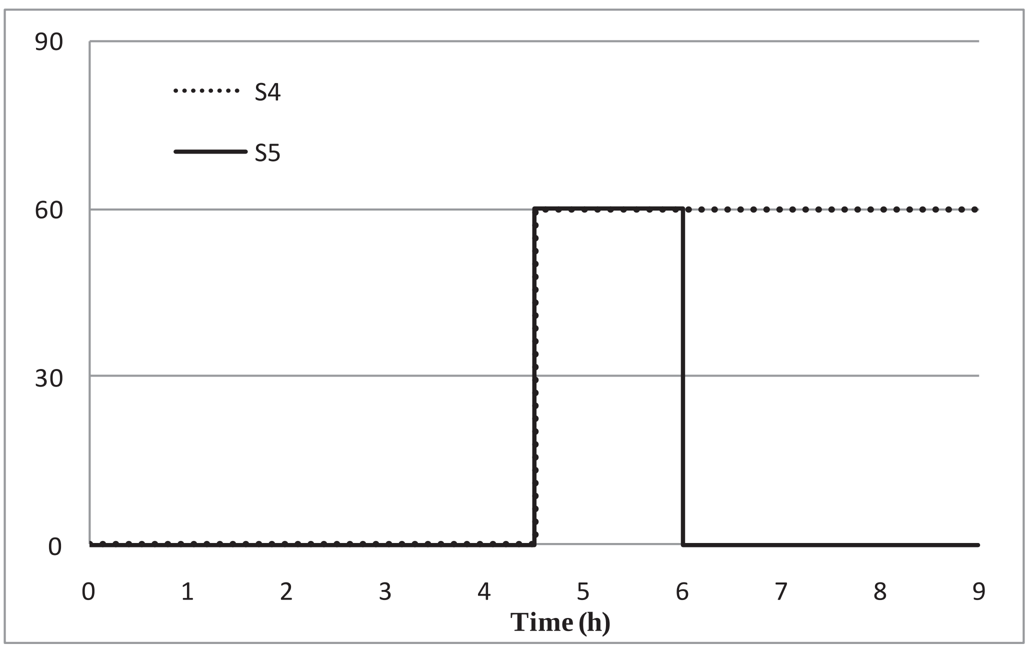

Figure 3.12 Inventory profiles of s4 and s5 for the schedule in Figure 3.11

For scenario B, the model [78] yields the schedule in Figure 3.11 with 180 mu of production. Figure 3.12 gives the corresponding inventory profiles of s4 and s5. This schedule is suboptimal, because the schedule in Figure 3.13 produces 210 mu. More importantly, the latter is feasible, because the inventories of s4 and s5 (Figure 3.14) satisfy their storage limits (60 mu for s4 and 60 mu for s5). As shown in Example 1, the model [78] does not use the shared resources efficiently and leads to suboptimal solutions. In this case, a production greater than 180 mu is not achievable, because the batch sizes are unnecessarily restricted by the storage capacity constraints as in Example 1.

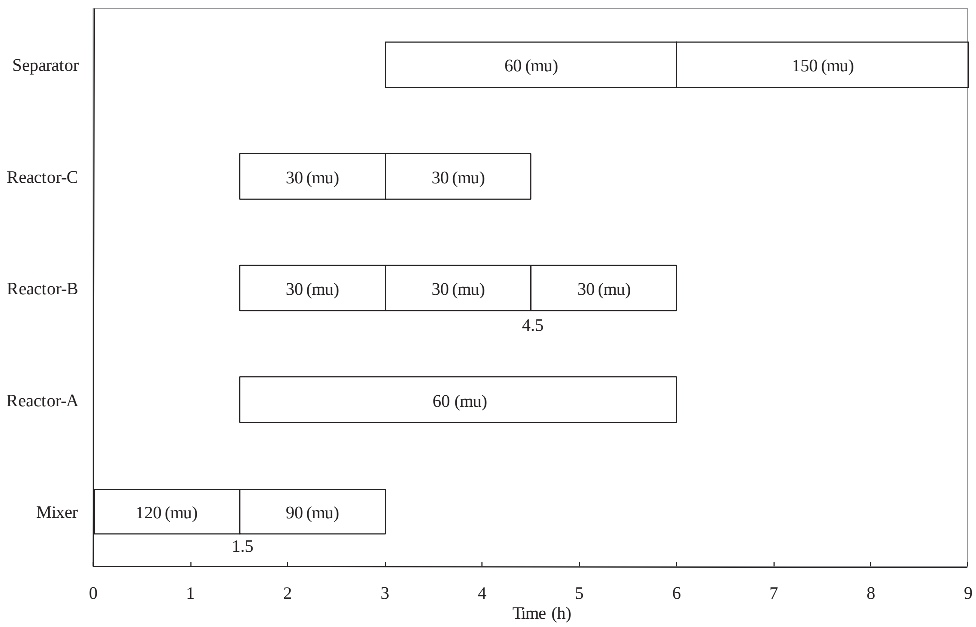

Figure 3.13 A feasible schedule for Scenario B of Example 2 obtained from Susarla et al. [206]

Examples 1 and 2 showed that the unit-specific event-based models that do not allow tasks to span over multiple events need refinement. Now, we show that the recent improved model of Shaik and Floudas [82] indeed addresses the possible concerns and solves both the examples successfully. Optimal solutions can also be obtained using other unit-specific event-based models [80, 81] that allow tasks to span over multiple events.

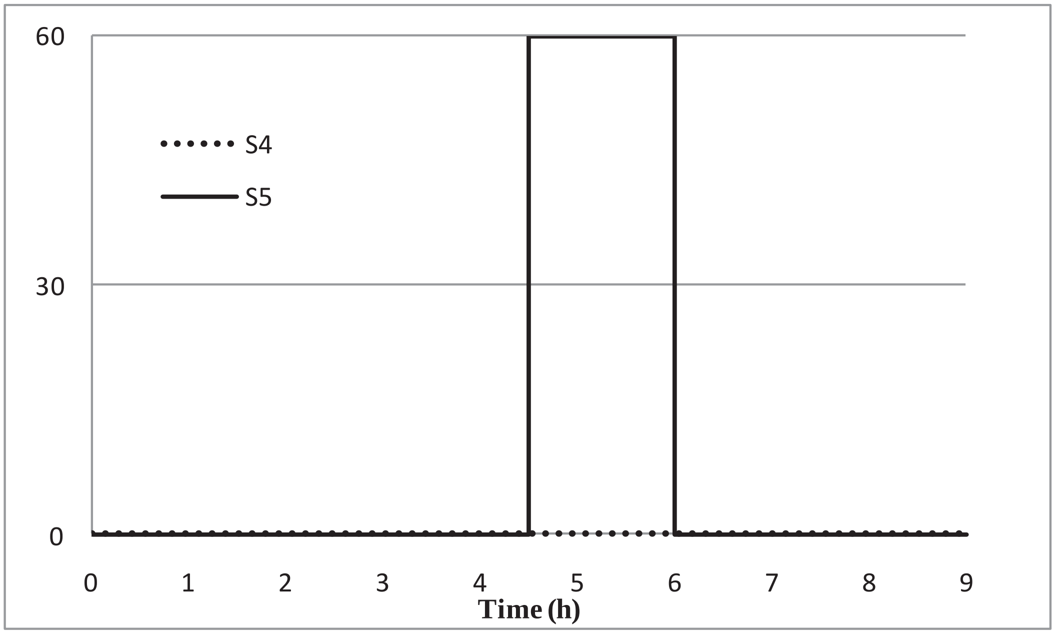

Figure 3.14 Inventory profiles of s4 and s5 for the schedule in Figure 3.13

3.5 Optimal solutions for Examples 1 and 2 using the unified model of Shaik and Floudas [82]

Shaik and Floudas [82] showed with an example that (especially) when there are shared resources such as utilities or even when there are no shared resources, it is a general requirement for all the unit-specific event-based (or unit-slot based) models, including the model of Ierapetritou and Floudas [74], its improved version as presented in Shaik et al. [85], and the RTN-based model of Shaik and Floudas [78], to allow tasks to span over multiple events in order to obtain optimal solutions. Otherwise, these models that allow tasks to occur over only single event may yield suboptimal solutions. Janak and Floudas [80] also had demonstrated this limitation. The first two examples on batch plants presented in this study fall under this category. Although, there are no explicit resources such as utilities, the finite intermediate storage in these two examples clearly acts as a shared resource over the parallel units.

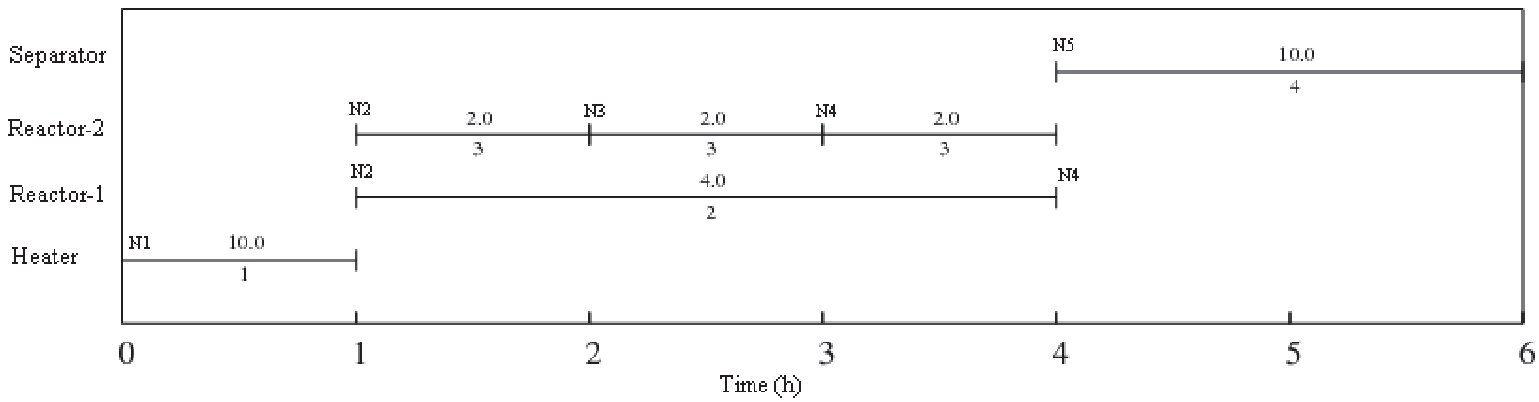

Figure 3.15 The Schedule for Scenario A of Example 1 from Shaik and Floudas [82]

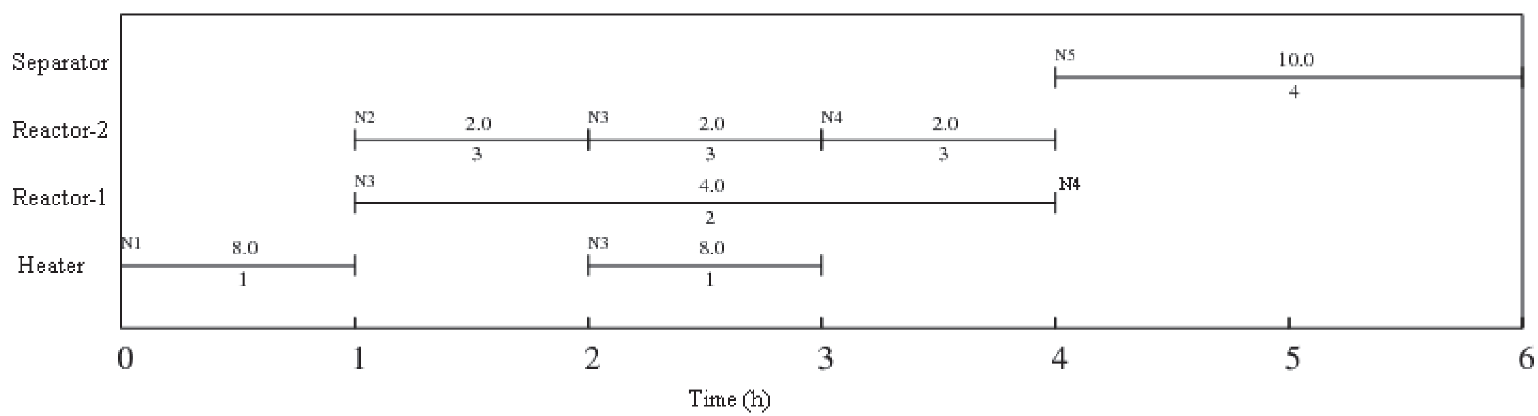

Figure 3.16 The Schedule for Scenario B of Example 1 from Shaik and Floudas [82]

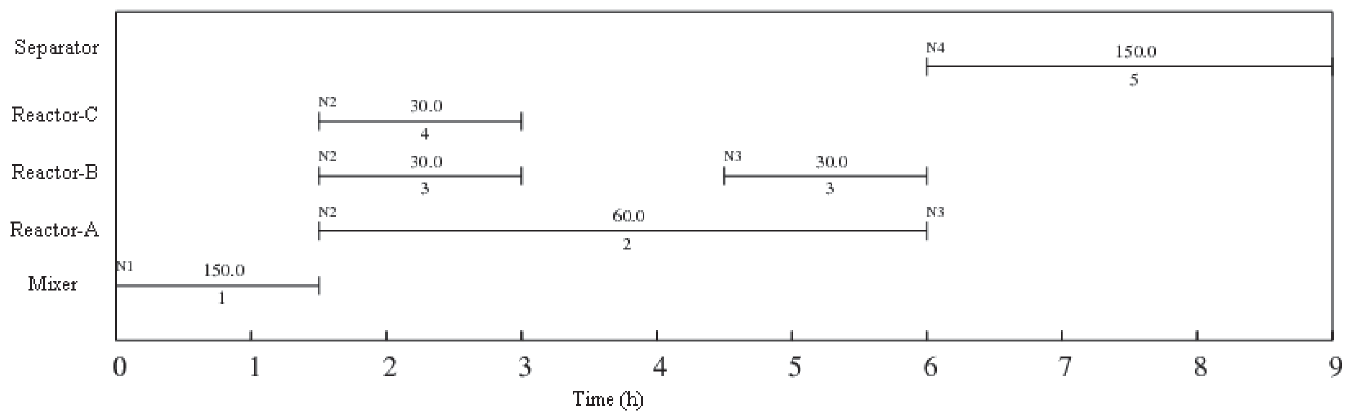

Figure 3.17 The Schedule for Scenario A of Example 2 from Shaik and Floudas [82]

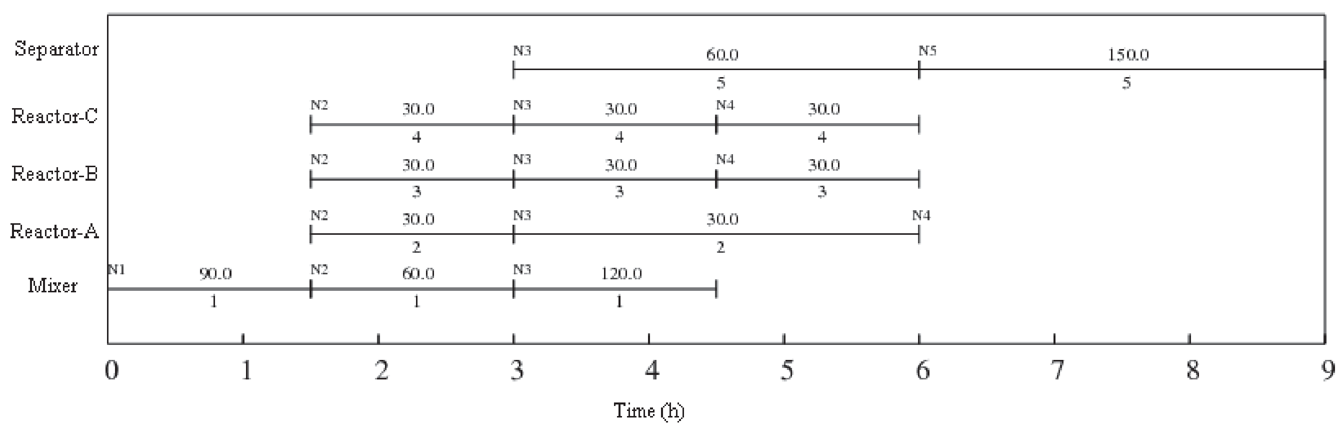

Figure 3.18 The Schedule for Scenario B of Example 2 from Shaik and Floudas [82]

We now present the solutions to these two examples using the model of Shaik and Floudas [82]. Since, both the examples of batch plants involve dedicated finite intermediate storage the first option is to consider additional sequencing constraints given in Shaik and Floudas [78, 82] to handle finite storage, although a more general approach would be to consider storage tasks explicitly. Shaik and Floudas [82] used an additional parameter, Δn, to control the number of events over which a task is allowed to span. This parameter also leads to compact problem size.

3.5.1 Example 1

For scenario A (fixed capacity), the unified model [82] finds the optimal solution successfully with a production of 10 kg using 5 events and Δn=2. The optimal schedule is given in Figure 3.15. For lower values of Δn the model yields trivial solution with no production, which also confirms that we need at allow tasks to span at least two events in order to obtain the optimal solution. In the optimal schedule of Figure 3.15, we can see that the task in Reactor-1 starts at event 'n2' and ends at event 'n4' due to limited storage.

For scenario B (variable capacity), the unified model [82] finds the optimal solution successfully with a production of 10 kg using 5 events and Δn=1. The optimal schedule is given in Figure 3.16. For Δn=0 (meaning tasks occur over single event), the model yields a sub-optimal schedule with a production of only 8 kg, which confirms the earlier results using Shaik and Floudas [78] that had this limitation. In the optimal schedule of Figure 3.16, we can see that the task in Reactor-1 starts at event 'n3' and ends at event 'n4' due to limited storage.

3.5.2 Example 2

Similar to example 1, this example also involves finite storage that leads to a shared resource over the parallel units. For scenario A (fixed capacity), the unified model [82] finds the optimal solution successfully with a production of 150 mu using 4 events and $\Delta n=1$. The optimal schedule is given in Figure 3.17, where it can be seen that the task in Reactor-A starts at event 'n2' and ends at event 'n3'. For $\Delta n=0$, the model yields a trivial schedule with no production, which confirms the earlier results using Shaik and Floudas [78].

For scenario B (variable capacity), the unified model [82] finds the optimal solution successfully with a production of 210 mu using 5 events and $\Delta n=2$. The optimal schedule is given in Figure 3.18, where it can be seen that the task in Reactor-A starts at event 'n2' processing 30 mu, continues over event 'n3' where the same amount 30 mu is shown, and finally ends at event 'n4'. For lower values of Δn the model yields a suboptimal solution with a production of 180 mu, which also confirms that we need at allow tasks to span at least two events in order to obtain the optimal solution. In both examples, the inventory levels in both event- and time-domains are found to be within the storage capacity limitations for both the scenarios.

3.6 Summary

We presented two examples to analyse the limitations of some unit-specific event-based models [78, 85] for scheduling multipurpose batch processes. Our study confirms that the examples involving batch plants, one reason for the limitations of these models is the allocation of only one event point for each task, which has also been previously demonstrated by Janak and Floudas [80] and Shaik and Floudas [82].

Thus, a general scheduling model should allow each task to span multiple event points (or slots) [70, 76, 80-82]. More importantly, we showed that the recent model of Shaik and Floudas [82] indeed addresses the limitations of previous models by allowing a task to span several event points and solves the first two examples on batch plants successfully. Additionally, the task sequencing constraints in these models may require further study to assure generality. We believe that these examples can serve as test problems for future scheduling models.

4 A NOVEL APPROACH TO SCHEDULING MULTIPURPOSE BATCH PLANTS USING UNIT-SLOTS[1, 2]

4.1 Introduction

The flexibility and versatility of batch plants in general and multipurpose batch plants (MBPs) in particular provide both opportunities and challenges for the manufacturer. MBPs employ a pool of equipment and resources to produce a slate of products with varying recipes and characteristics. Since the pool can be configured in a myriad of combinations and equipment and limited resources are shared among multiple products, scheduling the operation of MBPs is quite challenging and has received considerable attention in the literature. The allocation of equipment and resources and sequencing of various tasks over time are the main considerations in most scheduling problems. Thus, an effective approach for modeling time (or time representation) is of utmost importance in a MILP model. The substantial research effort over the past three decades has resulted in numerous mixed-integer linear programming (MILP) models.

Our specific goal in this chapter is to modify the model of Sundaramoorthy and Karimi [70] to use unit-slots instead of process-slots. In addition, we allow various

[1] Susarla, N., Li, J., & Karimi, I. (2010). A novel approach to scheduling multipurpose batch plants using unit slots. AIChE Journal, 56, 1859-1879.
[2] Susarla, N., Li, J., & Karimi, I. A. (2009). Unit-slots based short-term scheduling for multipurpose batch plants. Presented in PSE 2009, Salvador-Bahia-Brazil.

types of storage configurations [41, 46] and wait policies for material states. While we consider only "material" as a shared resource and "inventory balance" as a resource balance in this paper, we use "resource" and "resource balance" as generic terms in our discussion, because we believe that our approach is seamlessly extensible to other resources such as tools, instruments, parts, utilities, manpower, etc.

Given our discussion in Chapter 2, on the conceptual similarity between the models using unit-specific events and unit-slots, this chapter makes the following contributions. It presents a sound and systematic approach for handling shared resources in multi-grid models. It rationalizes and improves the approach that most unit-specific event-based models have used in the literature. Our approach can also be viewed as an extension and generalization of what Castro and Novais [83] did for addressing material transfers between processing units and inter-stage unlimited-capacity storage units in multi-stage batch plants with parallel units. Lastly and more importantly, it shows the limitation of the existing and presented multi-grid approaches for addressing shared resources.

4.2 Problem Statement

A MBP has J batch processing units ($j = 1, 2, \ldots, J$), performs I tasks ($i = 1, 2, \ldots, I$) involving (production or consumption) S material states ($s = 1, 2, \ldots, S$), and employs a dedicated storage s for each material state s. We describe the operation of MBP via the recipes [70] of various products, where a material state is any material (raw material, intermediate, waste, or final product) with distinct attributes and properties. To describe the multipurpose and specialty nature of units, we define $\mathbf{I}_j = (i \mid \text{unit } j \text{ can process task } i)$.

The MBP performs tasks in terms of individual batches. For each batch of task i on unit j in a MBP, we associate a batch size B_{ij}. We assume that this batch size requires a processing time of $\alpha_{ij} + \beta_{ij}B_{ij}$. Each batch will consume some material states in known proportions and produce other material states in some known proportions. We define a mass ratio (σ_{sij}) to quantify the actual amount of each material state s that a batch of task i on unit j may consume or produce. This is defined as follows.

$$\sigma_{sij} = \pm \frac{\text{Actual (not net) mass of material state } s \text{ consumed/produced by task } i \text{ on unit } j}{\text{Batch size (mass) of task } i \text{ on unit } j}$$

If task i on unit j consumes material state s, then $\sigma_{sij} < 0$; if it produces material state s, then $\sigma_{sij} > 0$; otherwise $\sigma_{sij} = 0$. Thus, the actual mass of material s associated with a batch size of B_{ij} is $|\sigma_{sij}|B_{ij}$. Note that the above mass ratio is unit-dependent. One may not need the mass ratio for a material state (e.g. a waste) in our model, if there is no need to monitor the inventory of such a material state. However, in several industrial scenarios, production schedules are constrained by the limited storage space for wastes or waste treatment capacity. Then, mass ratios will be needed for such waste materials in our formulation.

The storage of material states may involve various storage capacities and wait policies [41, 46]. These are unlimited intermediate storage (UIS), limited intermediate storage (LIS), no intermediate storage (NIS), unlimited wait (UW), limited wait (LW), and zero wait (ZW). Each task on a processing unit begins (ends) with the transfers of input (output) materials into (out of) that unit from (to) appropriate storage facilities.

With this, the scheduling problem addressed in this paper can be stated as:

Given:

1. Information on recipes, material states, tasks, mass ratios, etc.
2. J processing units, their suitable tasks, and limits on their batch sizes

3. *S* storage units, initial inventories, and limits on their holdups.

4. Wait policy for each material state *s*

5. Cost of or net revenue from each material state

Determine:

1. Tasks and their sequence and timings on each unit

2. Batch size of each task

3. Inventory profiles of all material states

Assuming:

1. Deterministic scenario with no batch/unit failures or operational interruptions.

2. Unit-to-unit transfers are instantaneous.

3. Setup or changeover times (if any) are lumped into batch processing times.

4. Batch processing time varies linearly with batch size.

5. All processing units can hold a batch temporarily before its start and after its end.

6. Direct unit-to-unit transfer of a material while bypassing the storage is allowed.

7. The storage of material states are the only shared resources.

8. Transfers of input materials for a batch may follow any sequence.

Allowing:

1. Transfers of input (output) materials into (out of) a unit for any batch need not be simultaneous.

Aiming for:

2. Maximum revenue from the plant for a given scheduling horizon [0, H], or

3. Minimum time (makespan) to produce specified demands (d_s, $s = 1, 2, \ldots, S$) of material states

Unless otherwise indicated, an index takes all its legitimate values in all the expressions or constraints in our formulation.

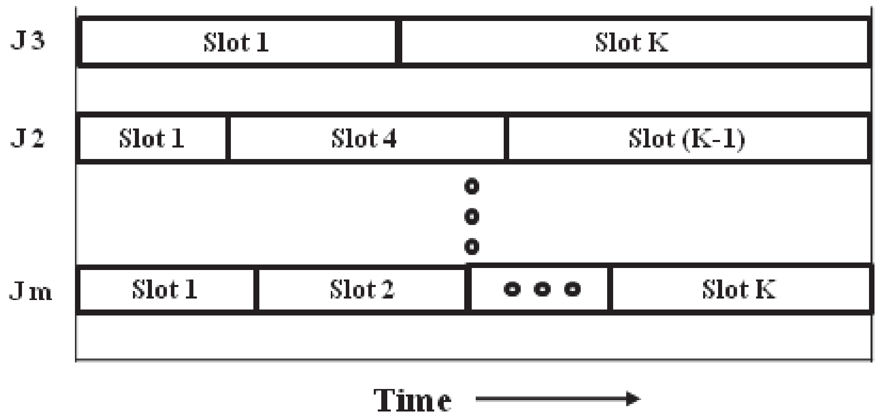

Figure 4.1 Design of unit-slots for our novel formulations

4.3 MILP Formulation

To model the schedule of activities on each unit j and the storage of each material state s during the scheduling horizon, we define K (k = 1, 2, …, K) contiguous slots (Figure 4.1) of unknown and arbitrary lengths. Let T_{jk} [k = 0, 1, 2, …, K; $T_{j0} \geq 0$; $T_{jK} \leq H$; $T_{jk} \geq T_{j(k-1)}$, $1 \leq k \leq K$] denote the end time of slot k on unit j. The time before slot 1 starts is slot 0 (k = 0). Thus, a slot k on unit j starts at $T_{j(k-1)}$, ends at T_{jk}, and has a length [T_{jk}– $T_{j(k-1)}$]. We use T_{sk} to denote the end times of slots on storage units. While each unit/storage has K slots, the slot end times (T_{jk} / T_{sk}) and thus slot lengths vary from unit to unit. By definition,

$$T_{j(k+1)} \geq T_{jk} \qquad 1 \leq j \leq J,\ 0 \leq k < K \qquad (4.1a)$$

$$T_{s(k+1)} \geq T_{sk} \qquad 1 \leq s \leq S,\ 0 \leq k < K \qquad (4.1b)$$

4.3.1 Tasks and Batches

We allocate the processing tasks to various slots on the processing units. Each task involves one batch, every slot must have a task, and the allocation of a task may span multiple slots. A batch of any task during [T_{jk}, $T_{j(k+1)}$] of a slot (k+1) on unit j involves three operations in the following order.

1. Unit j idles or receives input materials for the current batch during [T_{jk}, $T_{jk}+\delta_{j(k+1)}$], where $\delta_{jk} \geq 0$ ($1 \leq k \leq K$) is an unknown continuous variable, which also denotes the delay in the actual start of a task in slot k+1 of unit j. It may receive a material any time during this interval. Each material transfer into the unit is instantaneous, but the transfers need not be simultaneous and may follow any sequence. All the transfers required for the current batch end by $T_{jk}+\delta_{j(k+1)}$. If the unit is continuing a batch from the previous slot (k), then no idling or transfers can occur and $\delta_{j(k+1)} = 0$.

2. The unit processes (reacts, crystallizes, heats, etc.) the current batch. It begins the processing at $T_{jk}+\delta_{j(k+1)}$ and ends at $T_{j(k+1)}–\theta_{j(k+1)}$, where $\theta_{jk} \geq 0$ ($1 \leq k \leq K$) is another unknown continuous variable, which also denotes the early end of a task in slot k of unit j, such that $T_{j(k+1)}–\theta_{j(k+1)} \geq T_{jk}+\delta_{j(k+1)}$. This gives us the following that replaces eq. 1b.

$$T_{j(k+1)} \geq T_{jk} + \delta_{j(k+1)} + \theta_{j(k+1)} \qquad 1 \leq j \leq J,\ 0 \leq k < K \qquad (4.1c)$$

3. The unit idles or discharges the outputs from the current batch during [$T_{j(k+1)}–\theta_{j(k+1)}$, $T_{j(k+1)}$]. It may discharge an output material any time during this interval. The material transfers out of the unit are instantaneous, but need not be simultaneous[52-54]. All the required transfers end by $T_{j(k+1)}$. If the unit is continuing a batch into the next slot (k+2), then no idling or transfers can occur and $\theta_{j(k+1)} = 0$.

Next, we define one binary variable (ys_{ijk}) and two 0-1 continuous variables (ye_{ijk} and y_{ijk}) to denote respectively the start, end, and continuation of the allocation of a task (including the idling of a unit, which we define as task i = 0) on a unit-slot as follows,

$$ys_{ijk} = \begin{cases} 1 & \text{if task } i \text{ begins its allocation on unit } j \text{ in slot } (k+1) \\ 0 & \text{Otherwise} \end{cases}$$

$$1 \le j \le J,\ i = 0,\ i \in \mathbf{I}_j,\ 0 \le k < K$$

$$ye_{ijk} = \begin{cases} 1 & \text{if task } i \text{ ends its allocation on unit } j \text{ in slot } k \\ 0 & \text{Otherwise} \end{cases}$$

$$1 \le j \le J,\ i = 0,\ i \in \mathbf{I}_j,\ 1 \le k \le K$$

$$y_{ijk} = \begin{cases} 1 & \text{if task } i \text{ continues its allocation on unit } j \text{ from slot } k \text{ to } (k+1) \\ 0 & \text{Otherwise} \end{cases}$$

$$1 \le j \le J,\ i = 0,\ i \in \mathbf{I}_j,\ 0 \le k < K$$

ys_{ijk} refers to the start of a new allocation of task i from T_{jk}, y_{ijk} refers to the continuation of a current allocation of task i across T_{jk}, and ye_{ijk} refers to the end of a current allocation of task i at T_{jk}. If a unit j begins a new task in slot (k+1), then a new batch necessarily begins at T_{jk}. However, it is also possible that a unit ends a batch in slot k, and continues with a new batch of the same task again in slot (k+1). While y_{ijK} is undefined, y_{ij0} is known and fixed. If a task i is unfinished at time zero, and must continue, then $y_{ij0} = 1$, otherwise $y_{ij0} = 0$. Thus, we allow an unfinished batch at time zero to continue. However, we do not allow any unfinished batch at the end of scheduling horizon (H). In other words, all batches must end within the scheduling horizon. If a unit j has just ended a batch of i at time zero, then we set $ye_{ij0} = 1$ and all other ye_{ij0} as zero.

One of three things must happen at every T_{jk}. A task allocation may begin, a task allocation must continue, or the unit must idle. This gives us,

$$\sum_{i=0, i \in \mathbf{I}_j} (y_{ijk} + ys_{ijk}) = 1 \qquad 1 \le j \le J,\ 0 \le k < K \qquad (4.2)$$

Similarly, the allocation of a task i ends at T_{jk}, if and only if it starts/continues at $T_{j(k-1)}$, and does not continue across T_{jk}. That is,

$$ye_{ijk} = [y_{ij(k-1)} + ys_{ij(k-1)}] - y_{ijk} \qquad 1 \le j \le J,\ i = 0,\ i \in \mathbf{I}_j,\ 1 \le k < K \quad (4.3a)$$

$$ye_{ijK} = y_{ij(K-1)} + ys_{ij(K-1)} \qquad 1 \le j \le J,\ i = 0,\ i \in \mathbf{I}_j \qquad (4.3b)$$

4.3.2 Batch Sizes

Let $BI_{ijk} = B_{ij}^{L} ys_{ijk} + \Delta BI_{ijk}$ ($1 \le j \le J$, $i \in \mathbf{I}_j$, $0 \le k < K$), BO_{ijk} ($1 \le j \le J$, $i \in \mathbf{I}_j$, $1 \le k \le K$), and b_{ijk} ($1 \le j \le J$, $i \in \mathbf{I}_j$, $0 \le k < K$) respectively be the amounts of task i entering, exiting, and continuing at unit j at T_{jk} or the start of slot (k+1). Here, B_{ij}^{L} is the minimum required amount of task i on unit j. If a unit j is empty at the start of the horizon, then we set $b_{ij0} = 0$, otherwise we assign an appropriate nonzero value. We require that all units be empty ($b_{ijK} = 0$) at the end of the horizon.

First, the finite capacity of unit j demands that the amounts of task i entering, continuing, and exiting at T_{jk} not exceed the maximum allowable batch size (B_{ij}^{U}) of task i on unit j. Therefore, we have,

$$\Delta BI_{ijk} \le (B_{ij}^{U} - B_{ij}^{L}) ys_{ijk} \qquad 1 \le j \le J,\ i \in \mathbf{I}_j,\ 0 \le k < K \qquad (4.4a)$$

$$b_{ijk} \le B_{ij}^{U} y_{ijk} \qquad 1 \le j \le J,\ i \in \mathbf{I}_j,\ 1 \le k < K \qquad (4.4b)$$

$$BO_{ijk} \le B_{ij}^{U} ye_{ijk} \qquad 1 \le j \le J,\ i \in \mathbf{I}_j,\ 1 \le k \le K \qquad (4.4c)$$

Lastly, a balance on the amount of task i over slot k gives us,

$$BO_{ijk} = b_{ij(k-1)} + [B_{ij}^{L} ys_{ij(k-1)} + \Delta BI_{ij(k-1)}] - b_{ijk} \qquad 1 \le j \le J,\ i \in \mathbf{I}_j,\ 1 \le k \le K \qquad (4.5)$$

4.3.3 Operation Times

As described earlier, if the allocation of a task i continues across T_{jk}, then θ_{jk} for slot k and $\delta_{j(k+1)}$ for slot (k+1) must be zero. Thus,

$$\delta_{j(k+1)} \le H \sum_{i \in \mathbf{I}_j} ys_{ijk} \qquad 1 \le j \le J,\ 0 \le k < K \qquad (4.6a)$$

$$\theta_{jk} \le H \sum_{i \in \mathbf{I}_j} ye_{ijk} \qquad 1 \le j \le J,\ 1 \le k \le K \qquad (4.6b)$$

Next, let t_{jk} ($1 \le j \le J$, $0 \le k \le K$) denote the processing time remaining in completing the on-going batch on unit j at T_{jk}. If a batch has ended at or before time zero, then we set $t_{j0} = 0$, otherwise we assign an appropriate nonzero value. If a batch ends during slot k, then the remaining batch time must be zero at T_{jk}.

$$t_{jk} \le \sum_{i \in \mathbf{I}_j} (\alpha_{ij} y_{ijk} + \beta_{ij} b_{ijk}) \qquad 1 \le j \le J,\ 1 \le k < K \qquad (4.7a)$$

Using t_{jk}, we can compute the actual processing time (non-negative) of a batch during slot k as $t_{jk} + \sum_{i \in \mathbf{I}_j} [(\alpha_{ij} + \beta_{ij} B_{ij}^L) ys_{ijk} + \beta_{ij} \Delta BI_{ijk}] - t_{j(k+1)}$, where α_{ij} and β_{ij} ($1 \le j \le J$, $i \in \mathbf{I}_j$) are the parameters defining the linear dependence of batch processing time on batch size. Then, summing all the operation times, we obtain,

$$T_{j(k+1)} = T_{jk} + \delta_{j(k+1)} + t_{jk} + \sum_{i \in \mathbf{I}_j} [(\alpha_{ij} + \beta_{ij} B_{ij}^L) ys_{ijk} + \beta_{ij} \Delta BI_{ijk}] - t_{j(k+1)} + \theta_{j(k+1)}$$

$$1 \le j \le J,\ 0 \le k < K \qquad (4.7b)$$

4.3.4 Material Transfers and Inventory Balance

When a batch begins, it must use some materials from the storage tanks, and when it ends, it must transfer some materials to them. To this end, consider a unit j receiving or delivering a material state s in slot k. Suppose that this material flow occurs in slot k' on storage unit s. Three scenarios are possible: $k' < k$, $k' = k$, and $k' > k$. For $k' < k$, we can simply introduce additional slots on storage s to make $k' = k$. For $k' > k$, we can do the same on unit j. In other words, with no loss of generality, we demand that if a unit j is receiving or delivering a material to a storage s at any time, then the unit-slots corresponding to that time on both unit j and storage s must have the same index. This is a key step that avoids the binary variables defining the relative positions of checkpoints on different units as used by Lim and Karimi [63].

Now, consider the start of a new allocation of a batch of task i ($i \in \mathbf{I}_j$) in slot (k+1) of unit j. This batch will need materials from storage tanks s with $\sigma_{sij} < 0$, hence consider such a storage s transferring material s to unit j during slot (k+1) of unit j. As discussed earlier, we demand that this transfer must occur during [T_{jk}, $T_{jk}+\delta_{j(k+1)}$]. As argued in the previous paragraph, storage s must make this transfer during its own slot (k+1) or at time T_{sk} with no loss of generality. Since the transfer must occur between [T_{jk}, $T_{jk}+\delta_{j(k+1)}$], we write,

$$T_{sk} \geq T_{jk} - H[1 - \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} ys_{ijk}]$$

$$1 \leq j \leq J,\ s: \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} \sigma_{sij} < 0,\ s\text{: } I_s^U \text{ is limited, } 0 \leq k < K \quad (4.8a)$$

$$T_{sk} \leq T_{jk} + \delta_{j(k+1)} + H[1 - \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} ys_{ijk}]$$

$$1 \leq j \leq J,\ s: \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} \sigma_{sij} < 0,\ s\text{: } I_s^U \text{ is limited, } 0 \leq k < K \quad (4.8b)$$

Note that we do not use eqs. 4.8a and 4.8b for j and s, if j can never perform a task that consumes s, or s has an unlimited supply (e.g. raw materials) and can never experience a stock-out. $s: \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} \sigma_{sij} < 0$ ensures the former. The summation term in eq. 4.8 is to ensure that eq. 4.8 holds any time a task that consumes s on unit j begins at T_{jk}.

We now use a similar argument to address the transfer of material at the end of a batch on unit j. Consider the end of a batch of task i ($i \in \mathbf{I}_j$) during slot k on unit j. Each storage s with $\sigma_{sij} > 0$ will receive some material from this batch. As done before, we demand that this transfer must occur during [$T_{j(k+1)}-\theta_{j(k+1)}$, $T_{j(k+1)}$] on unit j and slot k

or at time T_{sk} on storage s. Therefore, we get,

$$T_{sk} \le T_{jk} + H[1 - \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} ye_{ijk}]$$

$$1 \le j \le J,\ s: \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} \sigma_{sij} > 0,\ 1 \le k < K,\ s\text{: } I_s^U \text{ is limited} \quad (4.9a)$$

$$T_{sk} \ge T_{jk} - \theta_{jk} - H[1 - \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} ye_{ijk}]$$

$$1 \le j \le J,\ s: \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} \sigma_{sij} > 0,\ 1 \le k < K,\ s\text{: } I_s^U \text{ is limited} \quad (4.9b)$$

Again, we do not use eqs. 4.9a and 4.9b for j and s, if j can never perform a task that produces s, or storage s is a final product or a waste with unlimited storage capacity, and can never face an overflow. $s: \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} \sigma_{sij} > 0$ ensures the former. The summation term in eq. 4.9 is due to the same reasons explained for eq. 4.8.

Eq. 4.9b assumes UW (Unlimited Wait) policy. To accommodate other wait policies such as LW (Limited Wait) and ZW (Zero Wait) for intermediate material states, we need the following.

$$T_{sk} \le T_{jk} - \theta_{jk} + \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} [w_{ij} ye_{ijk}] - H[1 - \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} ye_{ijk}]$$

$$1 \le j \le J,\ s: \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} \sigma_{sij} > 0,\ 1 \le k < K,\ s\text{: } I_s^U \text{ is LIS/LW} \quad (4.9c)$$

where, w_{ij} is the maximum time that unit j can hold a batch of task i after its completion of processing.

Based on our above discussion and eqs. 4.7-4.9, we have the following inventory balance for storage s.

$$I_{s1} = I_{s0} + \sum_{j} \sum_{i \in \mathbf{I}_s, \sigma_{sij} < 0} \sigma_{sij} [B_{ij}^L ys_{ij1} + \Delta BI_{ij1}] \qquad 1 \le s \le S \quad (4.10a)$$

$$I_{sk} = I_{s(k-1)} + \sum_{j} \sum_{i \in \mathbf{I}_s, \sigma_{sij} > 0} \sigma_{sij} BO_{ijk} + \sum_{j} \sum_{i \in \mathbf{I}_s, \sigma_{sij} < 0} \sigma_{sij} [B_{ij}^{L} ys_{ijk} + \Delta BI_{ijk}]$$

$$1 \le s \le S,\ 1 \le k < K \qquad (4.10b)$$

$$I_{sK} = I_{s(K-1)} + \sum_{j} \sum_{i \in \mathbf{I}_s, \sigma_{sij} > 0} \sigma_{sij} BO_{ijK} \qquad 1 \le s \le S \qquad (4.10c)$$

where, I_{sk} ($I_s^L \le I_{sk} \le I_s^U$) is the inventory of material s at T_{sk} or the end of slot k on storage s.

Note that temporary storage in processing units, instantaneous material transfers, and storage unit bypassing are key implicit assumptions in eqs. 4.7-4.10. Eq. 4.10 applies to the various storage capacities (LIS, UIS, and NIS) by setting I_s^U properly. However, we do not need eqs. 4.8a and 4.9a for the case of UIS.

Eqs. 4.8 and 4.9 are the key constraints that register material transfers chronologically on storage tanks. They do not exist in a formulation that uses process-slots. While process-slots are synchronized across all units irrespective of their relation to a material state, unit-slots are synchronized across only those units that are associated with a material state. The unit-slots in our formulation will be identical to process-slots in the worst-case scenario. In such a case, a formulation using process-slots will perform better (in terms of both computational time and relaxed MILP objective) than the one using unit-slots, because of the absence of eqs. 4.8 and 4.9.

We now argue that eqs. 4.8 and 4.9 guarantee a valid resource (specifically mass in this case) balance. A pivotal basis underlying this guarantee is our model's ability to allow tasks to span multiple slots. As discussed earlier, this allows us insert dummy slots freely as needed on any unit. This then allows us to assume with no loss of generality that if a unit j receives or delivers a material to a storage s at any time, then the unit-slots corresponding to that transfer must have the same index on both unit j and storage s. Now, eqs. 4.7 and 4.8 register exactly every resource exchange between a processing unit (*j*) and a resource on the time-grid of that resource (*s*). The timings of these exchanges are the endpoints (T_{sk}) of the slots on the time-grid of resource *s*, which are in the correct chronological order because of eq. 4.1. No resource exchange occurs at points other than T_{sk}. This proves that our resource balance is valid and correct.

Furthermore, since the various operations (transfer, processing, etc.) on each unit are allocated to their appropriate slots, and sequenced properly, the overall schedule will not have any infeasibility. Note that our model does not use any example-specific arguments to derive the timing constraints. Hence, it is as general as any model that uses global event points or process slots, as far as mass as a resource is concerned. However, in contrast, most models in the literature assume simultaneous (but instantaneous) transfers of materials. Since our model allows non-simultaneous (but instantaneous) transfers of materials, it is in fact more general than the models with global event points or process slots [70, 75, 76]. Indeed, we show later with an example that our model can give better optimal schedules than literature models [70, 76].

Castro et al. [77] and Gimenez et al. [207, 208] have also addressed the non-simultaneous transfers of materials. Castro et al. [77] defined explicit transfer tasks and assigned each material to a single transfer. In addition, they did not allow the temporary storage before the actual start of a task. The single-grid models of Gimenez et al. [207, 208] and our multi-grid model do not have such limitations. δ_{jk} and θ_{jk} in our model are very similar to the slack variables ($\tilde{T}_{j,n}^{LB}$ and $\tilde{T}_{j,n}^{EE}$) of Gimenez et al. [207, 208]. However, Gimenez et al. [207, 208] use several extra binary variables such as S_{jn}^{I} and S_{jn}^{O} to model the activity states of processing units, which our approach does not. Similarly, they use a binary variable Y_{ijn} to denote if a task *i* formally ends in unit *j* at T_n, which we do not. Therefore, our model uses much fewer binary variables. It detects activity states of the processing unit in an implicit manner, and is much simpler. However, Gimenez et al. [207, 208] also consider preventive maintenance, changeover or set-up times, and intermediate due-dates, which we do not in this work.

As mentioned earlier, the above approach for material transfers and inventory balances may be viewed as an extension of the one used by Castro and Novais [83] for scheduling multi-stage batch plants. However, a very critical distinction between the two is that Castro and Novais [83] do not allow a task to span multiple event-points, and arbitrarily assume that each transfer between a processing and a storage unit occurs at the same event-point across two different time-grids. In contrast, we provide a rationale for the latter in light of the former. This to us is a prerequisite for the validity of resource balance in multi-grid formulations. Apart from that consideration, our model allows non-simultaneous material transfers and applies to multi-purpose rather than multi-stage batch plants. The storage units are pooled across all units rather than being dedicated to the parallel units in the stages immediately upstream and downstream of each inter-stage storage. We consider all storage configurations (UIS, LIS, NIS) and wait policies (UW, LW, ZW) as compared to UIS/UW considered by Castro and Novais [83]. Finally, we use binary variables in eqs. 4.8 and 4.9 to enable selective synchronization across storage and processing units.

In addition to the above, our approach can also be contrasted with the one used by Janak et al. [81]. Janak et al. [81] defined an extra set of storage tasks and corresponding binary variables. In contrast, we do not have such binary variables. They also defined timing variables for the storage tasks. Their timing variables were essentially 3-index (task, implicit unit, event-point). In contrast, our timing variables are 2-index (storage unit, slot). They also used additional variables for the storage task amounts, and several extra constraints for capacity, duration, and sequencing to ensure correct material balance. In contrast, our approach is much simpler, more intuitive, and more general, and requires far fewer variables and constraints. We do not report our numerical evaluation of that model in this paper, because it took excessive computation times on almost all problems. Our observation is consistent with what has been commented by Shaik and Floudas [82].

4.3.5 Variable Bounds and Scheduling Objectives

Appropriate bounds on the variables can improve solution time. All variables are nonnegative in our formulation, and the upper bounds for continuous variables are $\Delta BI_{ijk} \le B_{ij}^U - B_{ij}^L$, $BO_{ijk} \le B_{ij}^U$, $b_{ijk} \le B_{ij}^U$, $T_{jk} \le H$, $T_{sk} \le H$, and $I_s^L \le I_{sk} \le I_s^U$. Note that we do not impose an upper bound on t_{jk} as done by Sundaramoorthy and Karimi [70], as it is bounded by eq. 4.6a. Also, we do not impose an upper limit on the inventory of a material state that does not have a zero-wait policy at the end of the last slot *K*, as that inventory level will depend on what may happen after the last slot K.

Three scheduling objectives are possible. Two used in the literature are makespan and revenue. The third is net profit. The revenue from production is given by:

$$R = \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i\in \mathbf{I}_j}\sum_{s,\sigma_{sij}>0} \sigma_{sij} v_s BO_{ijk} \tag{4.11}$$

where, υ_s is the price of material *s*.

In case of makespan minimization, *H* ceases to be a given parameter, and we need to satisfy a given demand (d_s) for each material *s*.

$$\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i\in \mathbf{I}_j}\sum_{s,\sigma_{sij}>0} \sigma_{sij} BO_{ijk} \ge d_s \tag{4.12}$$

In this case, we use the following to compute makespan, even though only the first would be sufficient.

$$MS \geq T_{jK} \tag{4.13a}$$

$$MS \geq T_{sK} \tag{4.13b}$$

Eqs. 4.11a and 4.11b used together seem to give faster solutions.

The third objective of net profit is a generalization of revenue. It includes all materials rather than just those that are sold.

$$NP = \sum_{k=1}^{K-1}\sum_{j=1}^{J}\sum_{i\in\mathbf{I}_j}\sum_{s:\sigma_{sij}<0}\sigma_{sij}v_s[B_{ij}^L ys_{ijk} + \Delta BI_{ijk}] + \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i\in\mathbf{I}_j}\sum_{s:\sigma_{sij}>0}\sigma_{sij}v_s BO_{ijk} \tag{4.14}$$

This completes our first model (SLK1, eqs. 4.1-4.9, 4.11 or 4.12-4.13 or 4.14) for scheduling MBP stated earlier. We now present a slight modification (SLK2) of SLK1.

4.3.6 Alternate Model (SLK2)

In contrast to SLK1, we define an additional 0-1 continuous variable as follows.

$$z_{jk} = \begin{cases} 1 & \text{if unit } j \text{ ends a batch within slot } k \\ 0 & \text{Otherwise} \end{cases} \qquad 1 \leq j \leq J,\ 0 \leq k < K$$

If a unit j is idle or has ended tasks at time zero, then z_{j0} = 1. Otherwise, z_{j0} = 0.

Clearly, ye_{ijk}, ys_{ijk} , and z_{jk} must satisfy,

$$z_{jk} = \sum_{i=0, i\in\mathbf{I}_j} ys_{ijk} \qquad 1 \leq j \leq J,\ 0 \leq k < K \tag{4.15a}$$

$$z_{jk} = \sum_{i=0, i\in\mathbf{I}_j} ye_{ijk} \qquad 1 \leq j \leq J,\ 1 \leq k < K \tag{4.15b}$$

For SLK2, we use the above in place of eq. 4.2. Now, we rewrite eqs. 4.4b-c as follows.

$$\sum_{i\in\mathbf{I}_j} b_{ijk} \leq B_j^U[1 - z_{jk}] \qquad 1 \leq j \leq J,\ 1 \leq k < K \tag{4.4d}$$

$$\sum_{i \in \mathbf{I}_j} BO_{ijk} \leq B_j^U z_{jk} \qquad 1 \leq j \leq J,\ 1 \leq k \leq K \qquad (4.4e)$$

where, $B_j^U = \max_{i \in \mathbf{I}_j} [B_{ij}^U]$.

With this, SLK2 comprises eqs. 4.1, 4.3, 4.4a, 4.4d, 4.4e, 4.5-4.9, and 4.11-4.15. Although SLK2 has more constraints than SLK1, it performs better, as we show later. Furthermore, this is in spite of the fact that eqs. 4.4b-c are intuitively tighter than eqs. 4.4d-e.

4.4 Numerical Evaluation

A fair and an unbiased comparison demands careful attention on many factors [209] such as hardware, operating system, and software. In our study, we used CPLEX 11/GAMS 22.8 [210] on a Dell precision PWS690 workstation with Intel® Xeon® 3 GHz CPU, 16 GB RAM, running Windows XP Professional x64 Edition. To solve our models for various examples, the first step is to program them in GAMS.

4.4.1 GAMS implementation

In GAMS, the MODEL statement defines a sequence of constraints in an optimization model. Most MILP users know very well a major pitfall that this step involves as far as the solution and comparison of MILP models are concerned. While the sequence in the MODEL statement does not affect model statistics such as the numbers of constraints, variables, and non-zeros, the reality is that it has a profound effect on the model solution time. A change in the sequence of the constraints changes the solution time for the same model. This is no news to a researcher working with MILPs. However, more significantly, this fact enables one to manipulate the sequence of constraints in order to obtain a better solution time for a given model. Ironically, to our knowledge, this issue has not been reported or discussed yet in the process scheduling literature on MILP models. Our extensive numerical experience with a variety of scheduling models in general, and those in this work in particular, shows that it is highly unlikely for a single sequence of constraints to be the most efficient for all test examples. Not only this, it is difficult to guarantee that a given sequence will be the most efficient for all instances of a specific example. The issue in our opinion is similar to the observation of Liu and Karimi [46] that it is difficult to find a single MILP model that performs the best on all examples. In our experience, the solution times can vary by an order of magnitude with a change in the order of constraints. Table 4.1 lists some of the many observed results from our numerical work. Clearly, this issue is extremely critical in comparing MILP models and can easily give unsound results. However, it is not clear, if a satisfactory resolution of this issue is possible. While it is certainly beyond the scope of this work, we strongly believe that this critical issue, in addition to those mentioned by Karimi et al. [211], must be taken care of in any numerical evaluation of MILP models. In our numerical evaluation, we have eliminated the effect of this factor. While we are unaware of any theoretical or heuristic guidelines for determining the optimal or best possible order/sequence of constraints, we believe that

Table 4.1 Effect of constraint sequence in GAMS models on solution times

Example	Objective	Demand (mu) /Horizon (h)	Model	CPU Time (s)	
				Seq-1	Seq-2
3	Max Revenue	H=12 h	SLK-2	781	564
	Max Revenue	(H=16 h)	SK	95.41	68.64
4	Max Revenue	(H=10 h)	SK	57.84	45.01
5	Min MS	d(s12)=100, d(s13)=200 mu	SLK-2	249	701
	Min MS	d(s12)=d(s13)=250 mu	SLK-2	9.7	15.6
	Min MS	d(s12)=d(s13)=250 mu	SK	107	2458

Seq-1 for SLK2 and SK are the same as the sequences reported in the paper.

Seq-2 for revenue maximization are:

Model SLK2 \10a-c, 1a-b, 9a-b, 8a-b, 11, 5, 4a, 4d, 4e, 7a-b, 15a-b, 3a-b\;

Model SK \2, 3, 9, 13, 5, 6, 10, 4, 11, 12, 14, 20\;

Seq-2 for makespan minimization are:

Model SLK2 \12, 15a-b, 3a-b, 5, 4a, 4d, 4e, 7a-b, 8b-a, 9b-a, 1a-b, 13a-b, 10a-c\;

Model SK \22, 2, 3, 9, 13, 5, 6, 10, 4, 11, 12, 14, 21\;

Table 4.2 Effect of solver options in GAMS models on solution times

Example	Objective	Demand (mu) /Horizon (h)	Model	CPU Time (*s*) for CPLEX Options			
				default	scaind = 1	reduce = 2	dpriind = 2
3	Max Revenue	H = 12	SLK-2	781	603	217	588
	Max Revenue	H = 16	SK	377	387	351	329
4	Min MS	d(s8) = d(s9) = 200	SLK-2	821	1364	720	786
5	Min MS	H = 12	SLK-1	422	579	267	91
	Min MS	d(s12) = 100, d(s13) = 200	SK	727	1265	724	481
	Min MS	d(s12) = 100, d(s13) = 200	MG	1930	2272	2806	359

every work must report the constraint sequences of all scheduling models that it tests. As a fair practice, in all MILP model comparisons, the literature models must be implemented with the same order of constraints as done in the articles that reported the models. If such sequences are not available, or the authors chose to use a different sequence, then the authors must report the appropriate sequences along with reasons for doing so. Later, we list the constraint sequences for all models tested in this work.

In addition to the above, several other factors have a significant impact on the computational performance of MILP formulations These are MIP solver, solver version [70], solver tuning options (Table 4.2), example-specific fixing of variables and parameters (e.g. Big-M values), uneven fixing of variables across models, solution iterations in search of the best, etc. For instance, fixing of variables based on example-specific information exists in some work [81, 82]. This can be an advantage for a model in which it is done, and disadvantage for the one in which it is not. A comparison, in which some variables are fixed in some models based on example-specific information, and not in others, can lead to unreliable assessment.

First, we solve a simple example to compare the recently published unit-specific event-based model [82] with our models (SLK-1 and SLK-2). This example highlights the need for allowing tasks to span all possible events or slots.

4.4.2 Example 1

This example consists of 4 tasks (*i*1-*i*4), 4 units (*j*1-*j*4), 6 material states (*s*1-*s*6), and 4 storages (*s*2-*s*5). Figure 4.2 shows the detailed recipe diagram, and Table 4.3-4.4 list the complete data. We assume a scheduling horizon of 10 h and maximize revenue.

Table 4.3 Batch size data for Examples 1-5

Task	Label i	Unit	Label j	α_{ij}	β_{ij}	$B_{ij}^L - B_{ij}^U$ (mu)
			Example 1			
Task 1	1	Unit 1	j 1	1.666	0.03335	0-40
Task 2	2	Unit 2	j 2	2.333	0.08335	0-20
Task 3	3	Unit 3	j 3	0.667	0.0666	0-5
Task 4	4	Unit 4	j 4	2.667	0.00833	0-40
			Example 2			
Task 1	1	Unit 1	Unit 1	1.666	0.0778	0-30
Task 2	2	Unit 2	Unit 2	2.333	0.0667	0-10
Task 3	3	Unit 3	Unit 3	0.669	0.0777	0-30
Task 4	4	Unit 3	Unit 3	0.667	0.033325	0-40
Task 5	5	Unit 2	Unit 2	1.332	0.0556	0-30
Task 6	6	Unit 1	Unit 1	1.5	0.025	0-20
			Example 3			
Task 1	1	Unit 1	Unit 1	1.333	0.01333	0-100
		Unit 2	Unit 2	1.333	0.01333	0-150
Task 2	2	Unit 3	Unit 3	1	0.005	0-200
Task 3	3	Unit 4	Unit 4	0.667	0.00445	0-150
		Unit 5	Unit 5	0.667	0.00445	0-150
			Example 4			
Heating	H	Heater	HR	0.667	0.00667	0-100
Reaction-1	R1	Reactor 1	RR1	1.334	0.02664	0-50
		Reactor 2	RR2	1.334	0.01665	0-80
Reaction-2	R2	Reactor 1	RR1	1.334	0.02664	0-50
		Reactor 2	RR2	1.334	0.01665	0-80
Reaction-3	R3	Reactor 1	RR1	0.667	0.01332	0-50
		Reactor 2	RR2	0.667	0.00833	0-80
Separation	S	Separator	SR	1.3342	0.00666	0-200
			Example 5			
Heating-1	H1	Heater	HR	0.667	0.00667	0-100
Heating-2	H2	Heater	HR	1.000	0.01	0-100
Reaction-1	R1	Reactor 1	RR1	1.333	0.01333	0-100
		Reactor 2	RR2	1.333	0.00889	0-150
Reaction-2	R2	Reactor 1	RR1	0.667	0.00667	0-100
		Reactor 2	RR2	0.667	0.00445	0-150
Reaction-3	R3	Reactor 1	RR1	1.333	0.0133	0-100
		Reactor 2	RR2	1.333	0.00889	0-150
Separation	S	Separator	SR	2.000	0.00667	0-300
Mixing	M	Mixer 1	MR1	1.333	0.00667	20-200
		Mixer 2	MR2	1.333	0.00667	20-200

Table 4.4 Storage capacities, initial inventories, and material prices for Examples 1–5

Material	Example 1		Example 2		Example 3		Example 4		Example 5	
s	Storage Capacity (mu) I_s^U	Initial Inventory (mu) I_s^0	Storage Capacity (mu) I_s^U	Initial Inventory (mu) I_s^0	Storage Capacity (mu) I_s^U	Initial Inventory (mu) I_s^0	Storage Capacity (mu) I_s^U	Initial Inventory (mu) I_s^0	Storage Capacity (mu) I_s^U	Initial Inventory (mu) I_s^0
1	UL	AA	UL	AA	UL	AA	UL	AA	UL	AA
2	10	0	UL	AA	200	0	UL	AA	UL	AA
3	15	0	UL	AA	250	0	UL	AA	100	0
4	10	0	10	0	UL	0	100	0	100	0
5	15	0	5	0	-	-	200	0	300	0
6	UL	0	10	0	-	-	150	0	150	50
7	-	-	UL	0	-	-	200	0	150	50
8	-	-	UL	0	-	-	UL	-	UL	AA
9	-	-	-	-	-	-	UL	-	150	0
10	-	-	-	-	-	-	-	-	150	0
11	-	-	-	-	-	-	-	-	UL	AA
12	-	-	-	-	-	-	-	-	UL	0
13	-	-	-	-	-	-	-	-	UL	0

UL = Unlimited ; AA = Available as and when required
Example 1: Price for s6 is 10 $/mu
Example 2: Price for s7 is 10 $/mu and s8 is 5 $/mu
Example 3: Price for s4 is 5 $/mu
Example 4: Price for s8 and s9 is 10 $/mu
Example 5: Price for s12 and s13 is 5 $/mu

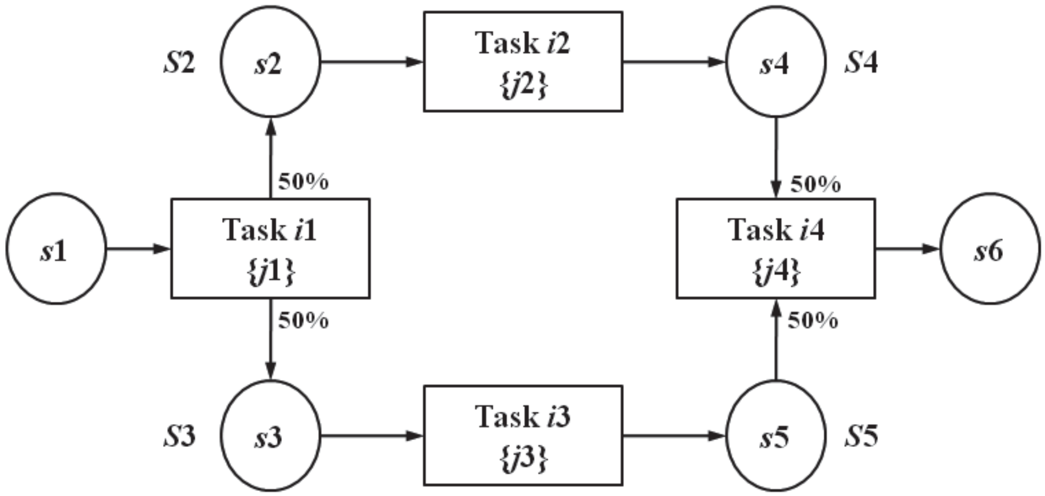

Figure 4.2 Recipe diagram for Example 1

First, we use the model (SF) of Shaik and Floudas [82]. We begin with n = 5 (five event-points) and examine the effect of increasing Δn [82] from 0 to 3. For Δn = 0, SF gives an objective of 200.12. Increasing Δn to 1 (Δn = 1) gives the same objective, so we increase n to 6. Now, SF gives the same objective of 200.12 for Δn = 0 and Δn = 1. Any further increase up to n = 15 gives the same solution for Δn = 0 and Δn = 1, so we conclude 200.12 as the best solution from SF. With SLK-1 and SLK-2, we get a solution of 300 for K = 6 and 400 with K = 7. Figure 4.3 gives that solution. Increasing K up to 15 does not improve the objective, so we conclude this as the best solution. Clearly, SF is unable to give the best solution using this common approach of increasing slots or event-points by one. The reason is that it has an additional layer of iteration, namely Δn. Therefore, to study SF further, we tried additional values for Δn. With n = 5 and Δn = 2, SF gives a solution of 300. Increasing Δn to 3 also gives the same solution. For n = 6 and Δn = 2, SF again gives a solution of 300. But with n = 6 and Δn = 3, SF gives the desired solution of 400. Thus, it is not clear how one should limit the number of event-points for tasks in SF, as it is not possible to know how many event-points a task may span in any given example. It is also clear that SF requires cascaded iterations as compared to our models.

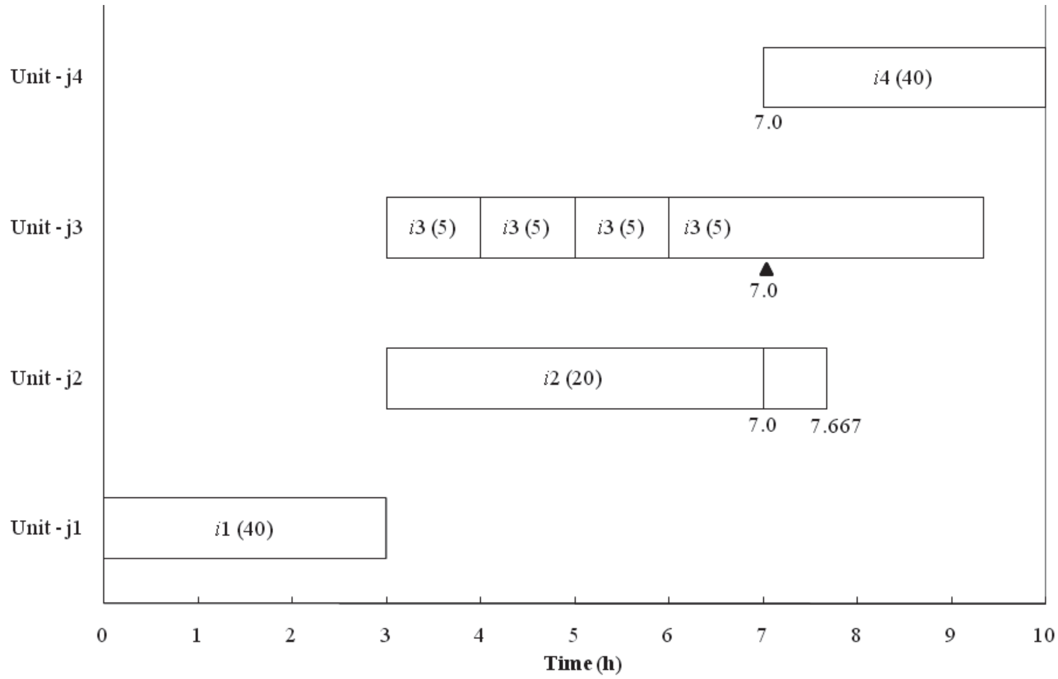

Figure 4.3 Schedule from SLK-2 for Example 1

Now, we present another example to show that SLKs can give better optimal schedules than those from the unit-specific event-based model of Shaik and Floudas (SF) [82], and the single-grid models of Maravelias and Grossmann (MG) [76] and Sundaramoorthy and Karimi (SK) [70]. This is simply because SLKs allow non-simultaneous material transfers, while others do not. Note that MG is indeed a reduced version of the model of Maravelias and Grossmann (MG) [76] by eliminating the constraints related to utilities/resources.

4.4.3 Example 2

This example involves 6 tasks (*i*1-*i*6), 3 units (*j*1-*j*3), 8 material states (*s*1-*s*8), 3 storages (*s*4-*s*6), and two final products (s7 and s8) via two production routes (R1 and R2). Figure 4.4 shows the recipe diagram with different arcs for R1 (solid) and R2 (dotted). Tables 4.3-4.4 list the complete data. We maximize revenue for a scheduling horizon of 6 h.

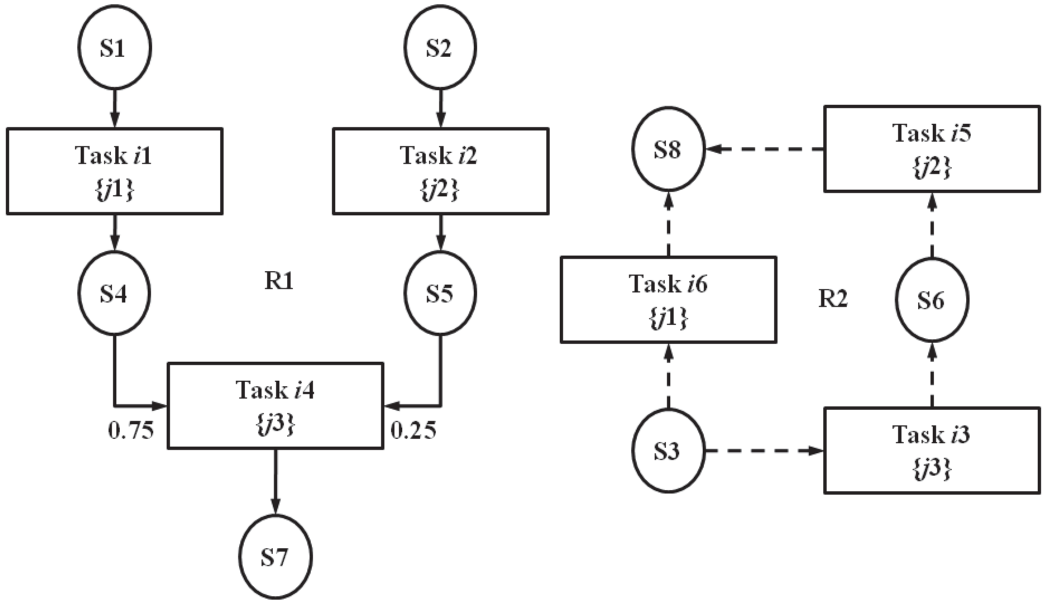

Figure 4.4 Recipe diagram for Example 2

MG, SK, and SF give an optimal solution of $560.1 for this example. In contrast, SLKs give a better optimal solution of $650. This is because the former do not allow non-simultaneous material transfers. For MG, SK, and SF assume that all materials required for each batch must be transferred simultaneously at a single point in time. Tasks *i*1 and *i*2 on *j*1 and *j*2 produce *s*4 and *s*5 respectively. *i*4 needs *s*4 and *s*5 in *j*3 to produce *s*7 (a final product). While *i*1 produces 30 kg of *s*4 in *j*1 at 4 h, *i*2 produces 10 kg of *s*5 in *j*2 at 3 h. Since MG, SK, and SF require that *s*4 and *s*5 must be transferred simultaneously to start *i*4 in *j*3, *s*5 has to wait 1 h (from 3 h to 4 h), before it is used for *i*4. Thus, *j*2 should either hold *s*5 during [3 h, 4 h], or transfer it to the storage at 3 h. Since S5 has a maximum capacity of 5 kg, it cannot store the 10 kg of *s*5, and *j*2 must hold *s*5 during [3 h, 4 h]. This forces *j*2 to be idle from 3 h to 4 h. SLKs on the other hand release *j*2 by allowing *s*5 to be transferred to *j*3 at 3 h. This enables *j*2 to have 3 h [3 h, 6 h] of production in SLKs instead of only 2 h [4 h, 6 h] in MG, SK, and SF. This forces MG, SK, and SF to use a lower batch size of *i*5 on *j*2 as compared to SLKs, and give an inferior solution. To ensure the validity of our results, we solved MG, SK, and

SF with up to 15 slots/events. The schedules for this example are shown in Figure 4.5 (from SLKs) and Figure 4.6 (from MG, SK, and SF).

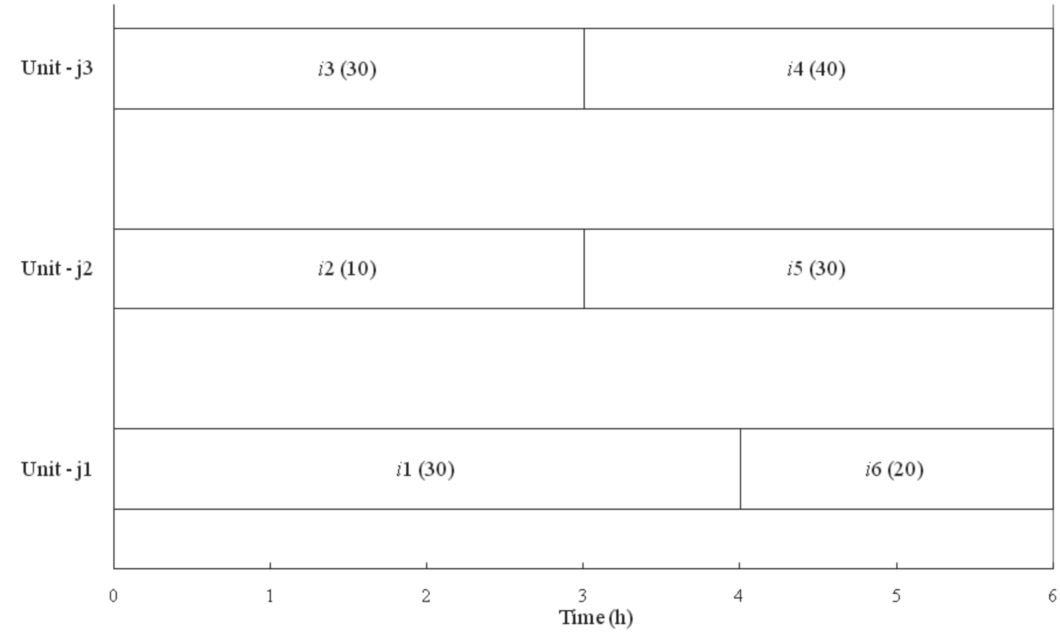

Figure 4.5 Schedule from SLKs for Example 2

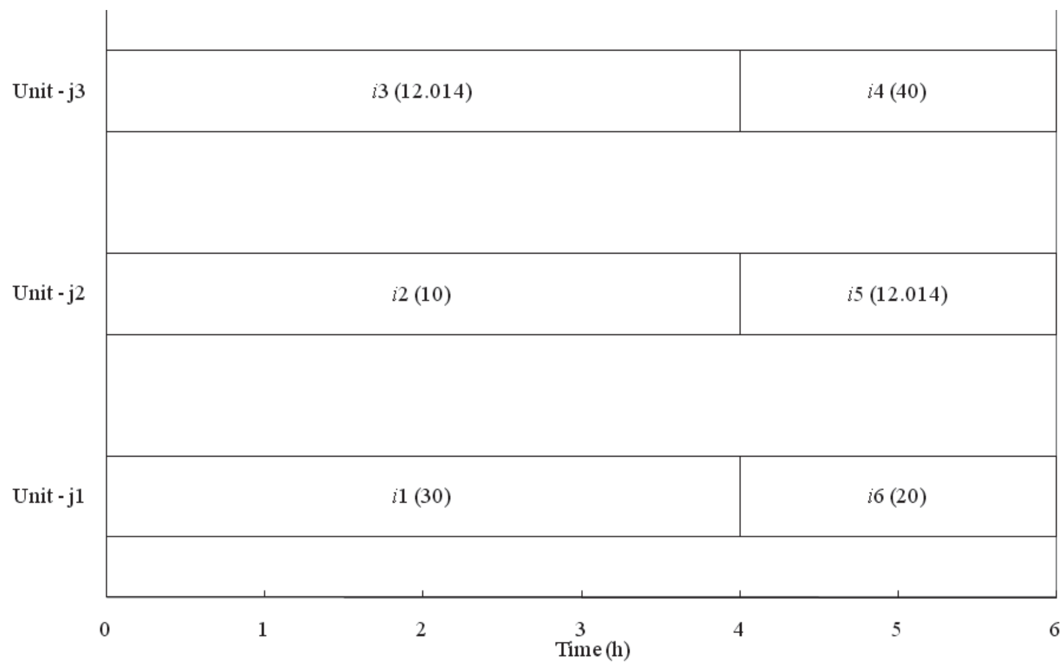

Figure 4.6 Schedule from MG, SK, and SF for Example 2

Now, we proceed to solve several cases of three more examples (Examples 2-4) from the literature to compare SLKs with MG, SK, and SF. However, since the latter do not allow non-simultaneous material transfers, comparing them would not be fair. Therefore, we reduce SLKs by forcing simultaneous material transfers to enable a fair comparison between the five models. We do this by setting $\delta_{jk} = 0$ and eliminating θ_{jk} from SLKs. The number of slots refers to the number of event points for SF and MG in all our subsequent discussion. In this work, we assumed UW policy for all examples and the following constraint sequences for the various models.

For revenue maximization:

Model SLK1 \1b, 1c, 2, 3a, 3b, 7a, M7b, 9a, M9b, M8b, 8a, 10a-c, 5, 4a, 4c, 4b, 11\;

Model SLK2 \1b, 1c, 15a, 15b, 3a, 3b, 7a, M7b, M8b, 8a, M9b, 9a, 10a-c, 5, 4a, 4d, 4e, 11\;

Model SK \2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 20\;

Model MG \11, 4, 5, 6, 12, 15-21, 3, 7, 23-30, 34-36\;

Model SF \1-17, 24, 25, 28-31, 33, 34\;

For makespan minimization:

Model SLK1 \1b, 1b, 2, 3a, 3b, 7a, M7b, 9a, M9b, 8a, 8Mb, 10a-c, 5, 4a, 4c, 4b, 12, 13a-b\;

Model SLK2 \1a-b, 15a-b, 3a-b, 7a, M7b, M8b, 8a, M9b, 9a, 10a-c, 5, 4a, 4d, 4e, 12, 13a-b\;

Model SK \22, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 21\;

Model MG \11, 4, 5, 6, 12, 15-21, 3, 7, 23-30, 34-36, 41\;

Model SF \1-17, 24, 25, 28-32\;

The equation numbers in the above are from Sundaramoorthy and Karimi [70] for SK, Maravelias and Grossmann [76] without resource constraints for MG, and Shaik and Floudas [82] for SF. The sequences correspond to the orders in which the equations are presented in the respective papers. Since the constraint sequences of MG, SK, and SF can also be "tuned", our numerical comparison is subject to the limitation of the above assumed sequences. Furthermore, as it was tedious to solve for all possible values of Δn in SF, we restricted to $\Delta n = 0$ and $\Delta n = 1$.

4.4.4 Example 3

This example [70] has been studied extensively in the literature. It involves 3 tasks (*i*1-*i*3), 5 units (*j*1-*j*5), 4 material states (*s*1-*s*4), and 2 storages (*S*2, *S*3). *j*1 & *j*2 can process *i*1, *j*3 can process *i*2, *j*4 and *j*5 can process *i*3. Figure 4.7 shows the recipe diagram. Tables 4.3-4.4 list the complete data.

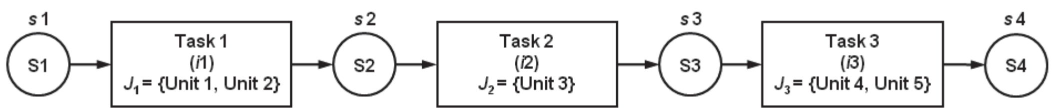

Figure 4.7 Recipe diagram for Example 3 [70]

First, consider revenue maximization for three scheduling horizons (3a: H = 10 h, 3b: 12 h, and 3c: 16 h). Table 4.5 gives the model and solution statistics. For this small problem, all models (SLK1, SLK2, SK, MG, and SF) expectedly have nearly similar statistics. For H = 10 h, SLK1 and SLK2 both need 6 slots (K = 7) to obtain the optimal solution of $2628.2 reported in the literature. However, if the same number of slots is the same, then single-grid models (SK and MG) give better RMIP objectives than multi-grid models (SLK-1, SLK-2, SF). This is mainly because the latter use several big-M constraints to synchronize the timings on different time grids.

Table 4.5 Model and solution statistics for Example 3

Model	K	H	CPU time (s)	Nodes	RMILP ($)	MILP ($)	Binary variables	Continuous variables	Constraints	Nonzeros	Relative Gap (%)
					Example 3a (H=10)						
SLK2	7	10	1.95	2764	4000.0	2628.2	60	359	491	1556	-
SLK1	7	10	3.20	6341	4000.0	2628.2	60	324	461	1496	-
SF											
($\Delta n = 0$)	6	10	0.17	285	3973.9	2628.2	30	119	209	639	-
($\Delta n = 1$)	6	10	1.33	1607	4000.0	2628.2	55	144	479	1539	-
SK	7	10	1.06	1090	3384.3	2628.2	60	316	300	1001	-
MG	7	10	0.88	770	3548.4	2628.2	70	309	846	2766	-
					Example 3b (H=12)						
SLK2	9	12	781	248340	4951.2	3463.6	80	467	657	2082	-
SLK1	9	12	1492	600476	4951.2	3463.6	80	422	617	2002	-
SF											
($\Delta n = 0$)	8	12	1.88	8136	4951.2	3463.6	40	157	281	865	-
($\Delta n = 1$)	8	12	585	254877	4951.2	3463.6	75	192	657	2117	-
SK	9	12	11.6	12480	4481.0	3463.6	80	416	408	1359	-
MG	9	12	10.3	32092	4563.8	3463.6	90	397	1084	3879	-
					Example 3c (H=16)						
SLK2	12	16	10000	1628804	6601.7	5038.1	110	629	906	2871	18
SLK1	12	16	10000	1294912	6601.7	5038.1	110	569	851	2761	19.1
SF											
($\Delta n = 0$)	11	16	113	484764	6601.7	5038.1	55	214	389	1204	-
($\Delta n = 1$)	11	16	10000	1727132	6601.7	5038.1	105	264	909	2984	15.87
SK	12	16	377	461037	6312.6	5038.1	110	566	570	1896	-
MG	12	16	2431	1974025	6332.8	5038.1	120	529	1441	5811	-
Model	K	H	CPU time (s)	Nodes	RMILP (h)	MILP (h)	Binary variables	Continuous variables	Constraints	Nonzeros	Relative Gap (%)
					Example 3d: d(s4) = 2000 mu						
SLK2	17	50	10000	328879	24.2	28.772	160	901	1330	4203	15.8
SLK1	17	50	10000	380619	24.2	28.772	160	816	1250	4043	15.8
SF											
($\Delta n = 0$)	16	50	10000	1093172	24.2	28.884	80	309	574	1783	6.6
($\Delta n = 1$)	16	50	10000	769022	24.2	28.772	155	384	1344	4443	15.8
SK	17	-	5403	3214852	24.72	28.772	160	816	843	2794	-
MG	17	50	10000	1210125	24.7	29.5	170	750	2045	9879	8.88
					Example 3e: d(s4) = 4000 mu						
SLK2	23	100	4944	1522250	48.5	56.432	220	1225	1828	5781	-
SLK1	23	100	10000	1880663	48.5	56.432	220	1110	1718	5561	1.89
SF											
($\Delta n = 0$)	22	100	34.6	42758	48.5	56.432	110	423	790	2461	-
($\Delta n = 1$)	22	100	8586	1957756	48.5	56.432	215	528	1860	6177	-
SK	26	-	10000	3115485	49.11	56.432	250	1266	1329	4405	6.04
MG	26	100	10000	562110	49.01	56.432	260	1146	3116	19212	10.25

For 3b (H = 12 h) and 3c (H = 16 h), the multi-grid models (SLK-1, SLK-2, and SF) solve slower than SK and MG. This is because multi-grid models are unable to reduce the number of slots as compared to single-grid models, and give poor RMIP values, which clearly shows their limitation. Both SK and MG give better RMIP objectives of \$4481 and \$4563.8 respectively, while the multi-grid models give \$4951.2. While SF solves faster for Δn = 0, one would need to solve it several times to get the best solution as discussed earlier. Thus, it is not feasible or fair to compare the solution times of SF with those of other models.

Figure 4.8 gives the schedule for Example 3b from SLK2. The rectangular blocks give task durations. Start/end times of batches along with slot numbers are shown under each block, and the batch sizes are indicated as labels.

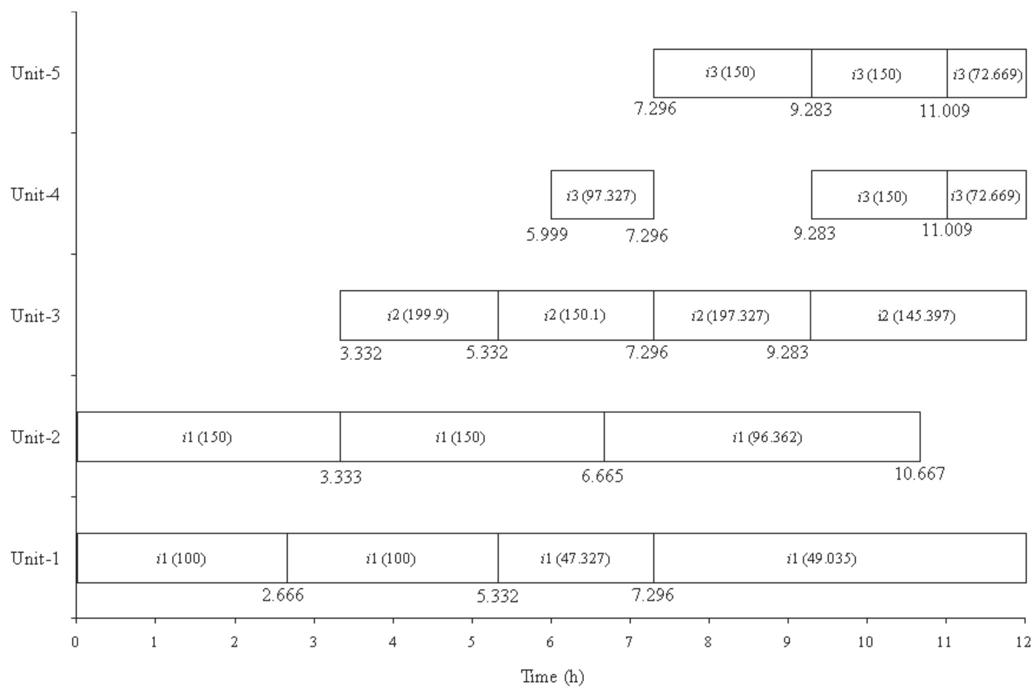

Figure 4.8 Maximum-revenue schedule from SLK-2 for Example 3b

For makespan minimization, we consider two cases with different product demands, d_4 = 2000 mu and 4000 mu. Table 4.5 lists the model and solution statistics. For d_4 = 2000 mu (Example 3d), both single- and multi-grid models need 16 slots (K =

17) to get the optimal solution of 28.772 h. However, single-grid models (SK and MG) give a better RMIP objective of \$24.7 as compared to \$24.2 for the multi-grid models (SLK-1, SLK-2, and SF). We allow a limit of 10000 CPU s for this example. SLK-1, SF (Δn = 0), and MG do not achieve the best solution of 28.772 h within 10000 CPU s. However, SK converges to 0% relative gap within 5403 s of CPU time. SLK-2 does not converge within 10000 CPU s, but attains the best solution. Here again, a single-grid model performs better than the multi-grid models due to the same number of slots.

For the more difficult case (Example 3e) of d_4 = 4000 mu, SLK-1, SLK-2, and SF require 23 slots (K = 24) to get the best makespan of 56.432 h. While SLK-2 converges in 4944 CPU s, SLK-1 does not converge in 10000 CPU s. SK and MG need 26 slots/events get the best solution, but after 10,000 CPU s. In this example, the reduction (Table 4.5) in the number of slots enables one multi-grid model (SLK-2) to outperform the single-grid models.

Note that the number of variables for our implementation of SF is slightly more than that reported in Shaik and Floudas [82], because we do not fix any variables (binary or continuous) based on specific problem details for the sake of a fair comparison. In contrast, Shaik and Floudas [82] fixed some variables for their model, but not for any other model.

4.4.5 Example 4

This example [34] is more complex than Example 3 and has been studied extensively in the literature. Figure 4.9 gives the recipe diagram. It involves 5 tasks (i1-i5), 4 processing units (HR, RR1, RR2, SR), 9 materials (s1-s9), and 4 storage units (S4-S7). HR can perform i1, RR1 and RR2 can perform i2-i4, and SR can perform i5. Tables 4.3-4.4 list the data.

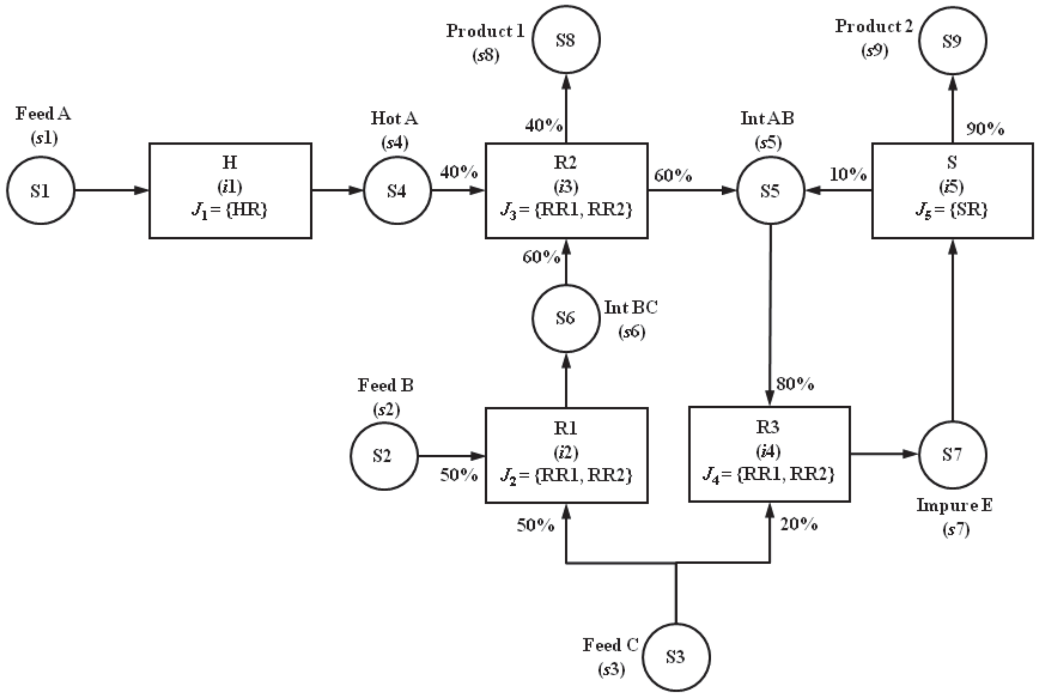

Figure 4.9 Recipe diagram for Example 4 [34]

For revenue maximization, we use three scheduling horizons (4a: H = 10 h, 4b: 12 h, and 4c: 16h). For Example 4a, Table 4.6 shows that multi-grid models require 1 fewer slot than single-grid models. However, in spite of this, the latter give a better RMIP objective of 2690.6 vs. 2730.7. This again may be due to the absence of big-M constraints in the latter. SLK-1 and SLK-2 solve faster (28.2 s and 12 s) than SK and MG (57.8 s and 126 s).

For Example 4b, SLKs perform exceptionally well and solve an order-of-magnitude faster than SK and MG. SLKs require 7 slots (K = 8) to get the optimal solution (Figure 4.10) of $2658.5. Surprisingly, even though SF is a multi-grid model, it needs one more event-point than those in SLKs. In other words, our proposed multi-grid approach is more effective in reducing slots than SF. In contrast, the single-grid models require 10 slots/events (K = 11), so our approach reduces three slots for this example. Consequently, SLK-1 and SLK-2 solve much faster (38 s for SLK-2 and 62.1 s for SLK-1 vs. 3330 s for SK and 9124.5 s for MG) and require fewer binary variables

(84 vs. 120 for SK and 160 for MG). SLKs also give the best RMIP objective of 3301.0 compared to 3350.5 for SF and 3343.4 for SK and MG.

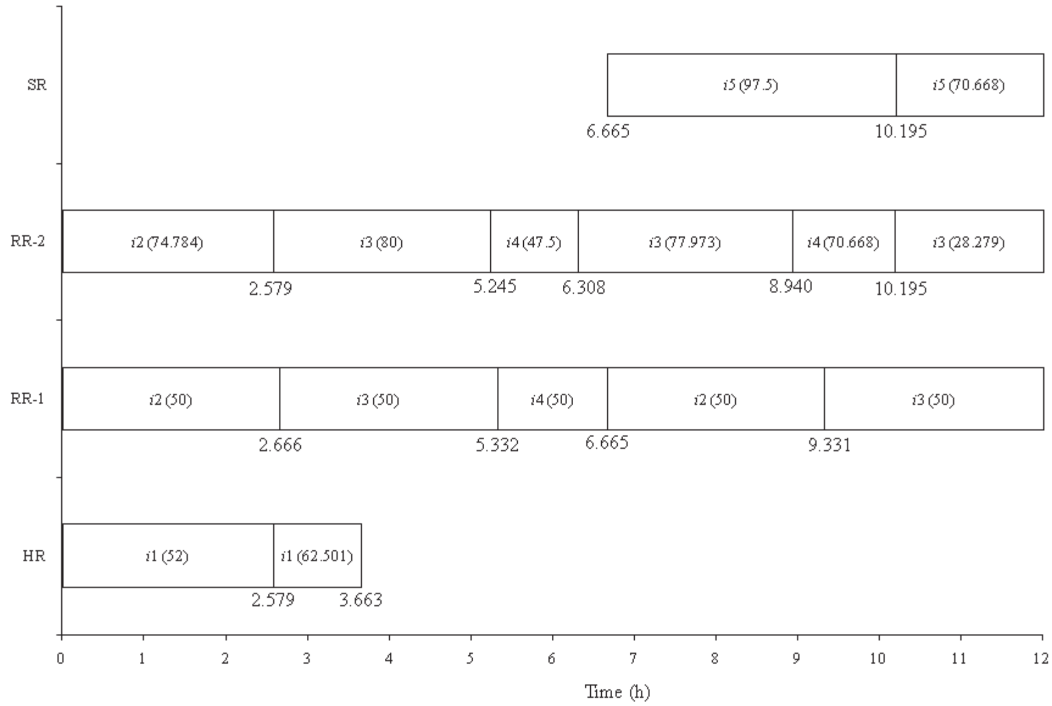

Figure 4.10 Maximum-revenue schedule from SLK-2 for Example 4b

For Example 4c, SLKs again perform quite well. They need only 8 slots (K = 9) compared to 9 slots/events (K = 10) for SF, SK, and MG to get the best solution of \$3738.38. This again confirms the ability of our approach to reduce slots, where SF does not. This reduction enables SLK-1 and SLK-2 to solve in only 30 s and 76.1 s respectively, while SK and MG require 156 s and 703 s respectively. The RMILP objective (\$4291.7.0) from SLKs is also better than that (\$4318.8) from SK, MG, and SF (\$4438.9). Figure 4.11 shows the optimal schedule from SLK-2.

For makespan minimization, we solve for two demands, d_8 = d_9 = 200 mu (Example 4d) and (d_8 = 500, d_9 = 400 mu) (Example 4e). Table 4.6 gives the model and solution statistics. For Example 4d, SLKs perform worse than SK and MG,

Table 4.6 Model and solution statistics for Examples 4

Model	K	H	CPU time (s)	Nodes	RMILP ($)	MILP ($)	Binary variables	Continuous variables	Constraints	Nonzeros	Relative Gap (%)
					Example 4a (H=10)						
SLK2	7	10	12.0	19043	2730.7	1962.7	72	449	645	2319	-
SLK1	7	10	28.2	44015	2730.7	1962.7	72	421	621	2315	-
SF											
($\Delta n = 0$)	6	10	3.19	7978	2730.7	1931.92†	48	208	439	1415	-
($\Delta n = 1$)	6	10	8.13	10915	2730.7	1962.7	88	248	847	3150	-
SK	8	10	57.8	65587	2690.6	1962.7	84	489	458	1686	-
MG	8	10	126	54753	2690.6	1962.7	112	617	1468	5464	-
					Example 4b (H=12)						
SLK2	8	12	38.0	58065	3301.0	2658.5	84	517	753	2710	-
SLK1	8	12	62.1	80093	3301.0	2658.5	84	485	725	2706	-
SF											
($\Delta n = 0$)	8	12	53.2	137217	3350.5	2658.5	64	274	597	1925	-
($\Delta n = 1$)	8	12	825	899959	3350.5	2658.5	120	330	1157	4342	-
SK	11	12	3330	2614949	3343.4	2658.5	120	687	665	2442	-
MG	11	12	9125	3174288	3343.4	2658.5	160	842	2020	8467	-
					Example 4c (H=16)						
SLK2	9	16	30.0	32531	4291.7	3738.38	96	585	861	3101	-
SLK1	9	16	76.1	62786	4291.7	3738.38	96	549	829	3097	-
SF											
($\Delta n = 0$)	9	16	140	315573	4438.9	3738.38	72	307	676	2180	-
($\Delta n = 1$)	9	16	3903	3554870	4438.9	3738.38	136	371	1312	4938	-
SK	10	16	156	96734	4318.8	3738.38	108	621	596	2190	-
MG	10	16	703	200592	4318.8	3738.38	144	767	1836	7410	-
Model	**K**	**H**	**CPU time (s)**	**Nodes**	**RMILP (h)**	**MILP (h)**	**Binary variables**	**Continuous variables**	**Constraints**	**Nonzeros**	**Relative Gap (%)**
					Example 4d: d(s8) & d(s9) = 200 mu						
SLK2	10	50	821	175107	18.7	19.34	108	658	983	3517	-
SLK1	10	50	1276	152939	18.7	19.34	108	618	947	3513	-
SF											
($\Delta n = 0$)	9	50	1.14	507	18.7	19.34	72	307	685	2199	-
($\Delta n = 1$)	9	50	1331	203829	18.7	19.34	136	371	1321	4957	-
SK	10	-	171	55349	18.7	19.34	108	621	604	2197	-
MG	10	50	314	36146	18.7	19.34	160	842	1962	7767	-
					Example 4e: d(s8) = 500 mu & d(s9) = 400 mu						
SLK2	22	100	10000	302206	47.4	47.6835	252	1474	2279	8209	0.64
SLK1	22	100	10000	260603	47.4	47.6835	252	1186	2195	8205	0.64
SF											
($\Delta n = 0$)	21	100	10000	986563	47.5	47.754	168	703	1633	5259	0.49
($\Delta n = 1$)	21	100	10000	340927	47.4	49.012	328	863	3181	12109	3.34
SK	23	-	10000	398979	48.78	49.05	264	1479	1501	5473	0.55
MG	23	100	10000	59431	48.78	49.05	368	1934	4471	26279	0.55

because the unit-slots are identical to process-slots. For Example 4e, although no model converges within 10000 CPU s, SLKs and SF need one slot less (*K* = 22) than SK and MG (*K* = 23). However, SLKs attain a better solution of $47.6835 than $47.75 for SF and $49.05 for SK and MG.

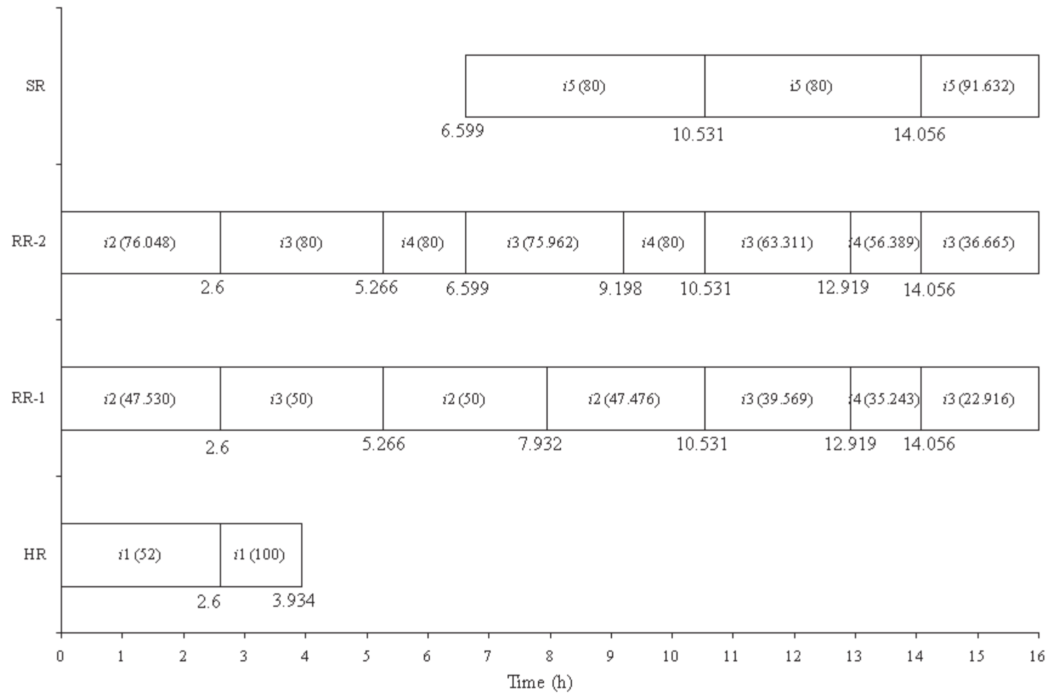

Figure 4.11 Maximum-revenue schedule from SLK-2 for Example 4c

4.4.6 Example 5

This example [70] (Figure 4.12) involves 6 processing units (HR, RR1, RR2, SR, MR1, and MR2), 7 tasks (*i*1-*i*7), 13 material states (*s*1-*s*13), and 7 storage units (*S*3-*S*7, *S*9, and *S*10). Relatively, this is a more complex problem, and hence used often in the literature. It embodies many common features of an MBP such as units performing multiple tasks, multiple units suitable for a task, and dedicated units for specific tasks. It also assumes non-zero initial inventories for *s*6 and *s*7, and recycles *s*4.

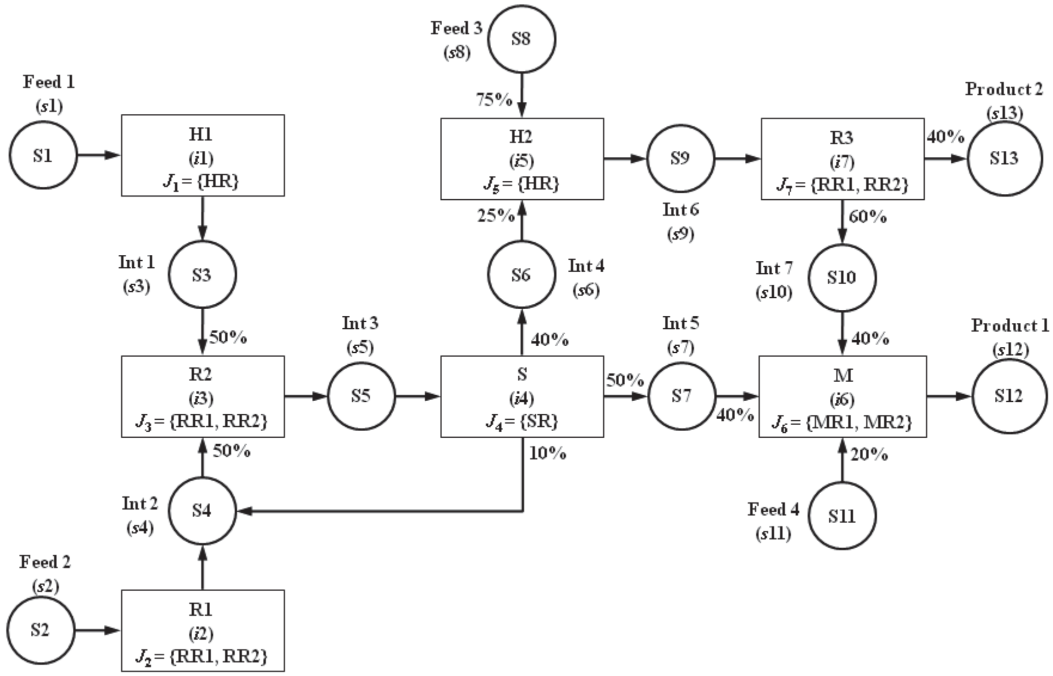

Figure 4.12 Recipe diagram for Example 5 [70]

For revenue maximization, we use four scheduling horizons; Example 5a: H = 8 h, Example 5b: H = 10 h, Example 5c: H = 12 h, and Example 5d: H = 16 h. From Table 4.7, all models require six slots/events (K = 7) for H = 8. Again, single-grid models perform better. For Example 5b (H = 10), although no model converges in 10000 CPU s, SLKs and SF require fewer slots (9 vs. 10) than SK and MG. Also, SLKs and SF obtain a solution of $2337.36 in 10000 s compared to $2260.9 for SK and $2137.1 for MG. For Example 5c (H = 12), SLKs and SF again require fewer slots/events (8 vs. 9) than SK and MG. The RMILP objectives are also better ($3465.6 vs. $3867.3), and SLK-2 is significantly faster (10.1 s vs. 95.4 s for SK and 296 s for MG). SLK-1 takes a relatively long time (> 400 s) to converge. For H = 16 h, although no model converges within 10000 CPU s, SLK-2 and SF attain a better objective ($4241.5 vs. $4240.83 for others).

For makespan minimization, we consider Example 5e: d_{12} = 100 and d_{13} = 200 mu, and Example 5f: d_{12} = d_{13} = 250 mu. For Example 5e, SLKs use up to three fewer

Table 4.7 Model and solution statistics for Examples 5

Model	K	H	CPU time (s)	Nodes	RMILP ($)	MILP ($)	Binary variables	Continuous variables	Constraints	Nonzeros	Relative Gap (%)
					Example 5a (H=8)						
SLK2	7	8	284	388832	2751.0	1583.4	102	655	1070	3671	-
SLK1	7	8	2659	3674795	2751.0	1583.4	102	613	1034	3654	-
SF											
($\Delta n = 0$)	6	8	9.37	18578	2751.0	1583.4	66	290	716	2169	-
($\Delta n = 1$)	6	8	56.8	80223	2751.0	1583.4	121	345	1336	4628	-
SK	7	8	45.5	39305	2560.6	1583.4	102	595	728	2207	-
MG	7	8	81.2	55146	2560.6	1583.4	154	806	1893	6630	-
					Example 5b (H=10)						
SLK2	9	10	10000	872204	3618.6	2337.36	136	853	1428	4907	5.07
SLK1	9	10	10000	711493	3618.6	2337.36	136	799	1380	4886	11.6
SF											
($\Delta n = 0$)	8	10	115	1160988	3618.6	2292.5†	88	382	884	2847	-
($\Delta n = 1$)	8	10	10000	2256858	3618.6	2337.36	165	459	1658	6187	1.68
SK	10	10	10000	2666604	3473.9	2260.9†	153	874	1121	3380	10.7
MG	10	10	10000	1415516	3473.9	2137.1†	220	1151	2691	10710	23.9
					Example 5c (H=12)						
SLK2	8	12	10.1	3096	3465.6	3041.3	119	754	1249	4289	-
SLK1	8	12	422	236847	3465.6	3041.3	119	706	1207	4270	-
SF											
($\Delta n = 0$)	7	12	2.11	1262	3465.6	3041.3	77	336	844	2560	-
($\Delta n = 1$)	7	12	4.44	881	3465.6	3041.3	143	402	1579	5505	-
SK	9	12	95.4	43951	3867.3	3041.3	136	781	990	2989	-
MG	9	12	296	71877	3867.3	3041.3	198	1036	2425	9269	-
					Example 5d (H=16)						
SLK2	11	16	10000	705090	5225.9	4241.5	170	1051	1786	6143	0.29
SLK1	11	16	10000	3042257	5225.9	4237.61	170	985	1726	6118	1.64
SF											
($\Delta n = 0$)	10	16	1475	515144	5225.9	4241.5	110	474	1228	3733	-
($\Delta n = 1$)	10	16	10000	1511596	5225.9	4241.5	209	573	2311	8136	0.01
SK	11	16	1687	283938	5125.9	4240.83	170	967	1252	3771	-
MG	11	16	10000	739728	5125.9	4185.24	242	1266	2957	12232	1.69
Model	**K**	**H**	**CPU time (s)**	**Nodes**	**RMILP (h)**	**MILP (h)**	**Binary variables**	**Continuous variables**	**Constraints**	**Nonzeros**	**Relative Gap (%)**
					Example 5e: d(s12) = 100 mu & d(s13) = 200 mu						
SLK2	9	50	249	45218	11.3	13.367	136	859	1448	4944	-
SLK1	9	50	2538	502461	11.3	13.367	136	805	1400	4923	-
SF											
($\Delta n = 0$)	9	50	70.7	43967	11.3	13.367	99	428	1112	3369	-
($\Delta n = 1$)	9	50	1867	770883	11.3	13.367	187	516	2079	7286	-
SK	11	-	727	233210	11.417	13.367	170	967	1264	3782	-
MG	11	50	1930	162982	11.417	13.367	242	1267	2970	12427	-
					Example 5f: d(s12) & d(s13) = 250 mu						
SLK2	11	100	9.70	820	14.3	17.025	170	1057	1806	6180	-
SLK1	11	100	6.34	663	14.3	17.025	170	991	1746	6155	-
SF											
($\Delta n = 0$)	10	100	5.38	2236	14.3	17.199	110	474	1240	3760	-
($\Delta n = 1$)	10	100	7.30	683	14.3	17.025	209	573	2323	8163	-
SK	12	-	107	12992	15.001	17.306	187	1060	1395	4173	-
	13	-	387	29981	14.920	17.306	204	1153	1526	4564	-
MG	12	100	247	14683	15.001	17.306	264	1382	3236	14047	-
	13	100	6798	244438	14.920	17.306	286	1497	3502	15748	-

slots (8 vs. 9 for SF and 8 vs. 11 for SK and MG). SLK-2 is significantly faster than SK and MG (249 s vs. 727 s for SK and 1930 s for MG). Figure 4.13 gives the detailed schedule. For Example 5f also, SLKs outperform SK and MG significantly as seen in Table 4.7.

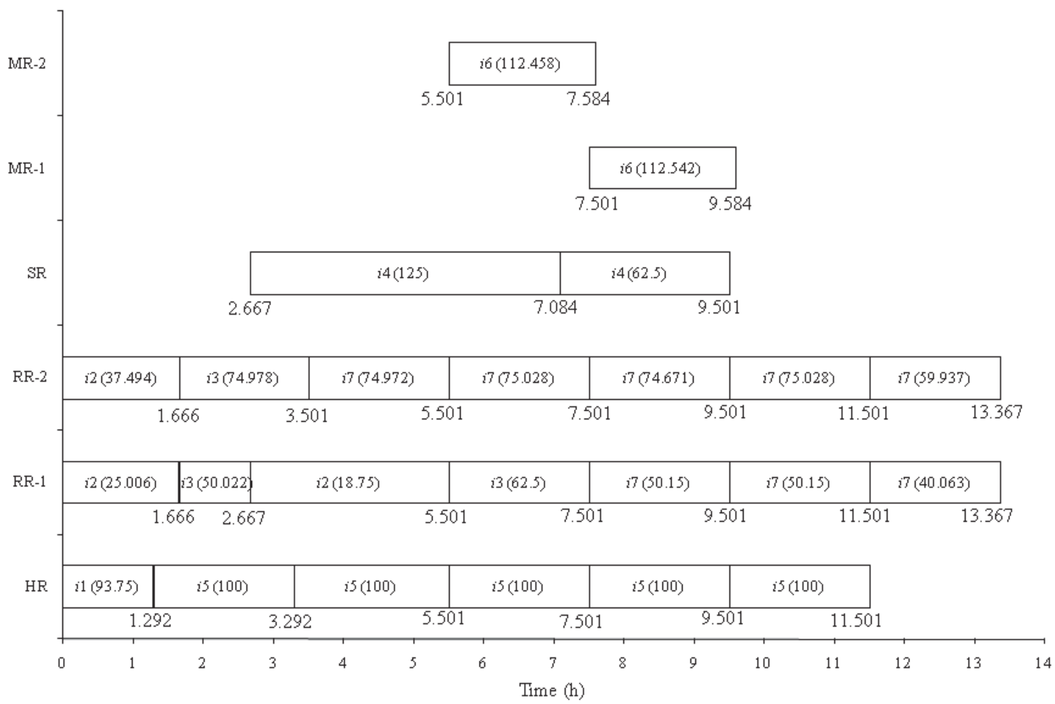

Figure 4.13 Minimum-makespan schedule from SLK-2 for Example 5e

4.5 Remarks

Our numerical evaluation demonstrates that our unit-slot models perform much better than some general models in the literature for both revenue maximization and makespan minimization except where unit-slots are identical to process-slots. Although SLK-1 performs nearly the same as SLK-2, SLK-2 seems to be more consistent across the limited problems considered in this work. Our models are much simpler in implementation and do not require cascaded iterations like SF. Additionally, as seen in some examples, our models require even fewer slots than the unit-specific event-based models (SF).

The main difference between our models (SLKs) and SK is of course the use of unit-slots and consequent additional variables (T_{jk} and T_{sk}) and constraints (eqs. 4.7 and 4.8) for synchronizing transfer timings between processing units and storages. However, some additional differences are worth noting.

1. In contrast to SK, SLKs do not use the slot lengths (SL_{jk}) as variables. This eliminates the constraint that forces the sum of slot lengths to be less than H.
2. Instead of writing eq. 4.5 for b_{ijk} as in SK and SLKs, we experimented with an aggregated form as follows.

   $$b_{jk} = b_{j(k-1)} + \sum_{i \in \mathbf{I}_j, i>0} (B_{ij}^L ys_{ijk} + \Delta BI_{ijk}) + \sum_{i \in \mathbf{I}_j, i>0} BO_{ijk}$$

3. Our motivation for trying the above form was to reduce the numbers of variables and constraints. However, we discovered that the above equation leads to a poor relaxation.
4. In contrast to SK, we express BI_{ijk} as $BI_{ijk} = B_{ij}^L ys_{ijk} + \Delta BI_{ijk}$ to reduce the number of constraints. Furthermore, eqs. 4.4b & 4.4c also eliminate several constraints that SK uses.
5. We have introduced an alternate objective of profit in addition to revenue and makespan, which are well known in the literature.
6. We have also generalized the concept of stoichiometric coefficient to Mass Ratio (σ_{sij}), which captures unit dependency and enables a proper mass balance on materials involved in a batch.
7. The use of eq. 4.2 is also novel compared to SK. SLK-2 is assumed to have a tighter relaxation with more variables and constraints. Also, this is why SLK-2 seems to perform slightly better than SLK-1 in spite of SLK-1 using fewer variables and constraints.

4.6 Summary

We proposed a novel continuous-time formulation for scheduling multipurpose batch plants. Its major contributions are a fool-proof and novel use of unit-slots in managing shared resources such as materials and the flexibility to allow non-simultaneous transfers of materials into a batch. It gives rational and logical arguments and constraints for resource balance using unit-slots. Similar to the unit-specific event-based models, it does not use the extra binary variables used by Lim and Karimi [63]. For some literature examples, our approach needs fewer slots/events than both single- and multi-grid models that exist in the literature. This enables significant reductions in solution times and model size, and yields tighter RMIP values. However, in problems where unit-slots are identical to process slots, the performance of multi-grid models is worse than single-grid models. Thus, it demonstrates the limitation of the current multi-grid models. Lastly, this work highlights the importance of constraint sequencing in GAMS implementation for evaluating MILP-based scheduling models fairly. While this particular work has not addressed all the other features of MBPs in the literature such as changeover times, semi-continuous processes, resources other than materials, etc. the proposed approach is readily extendible and are explored in the next chapter.

5 RESOURCE CONSTRAINED SHORT-TERM SCHEDULING FOR MULTIPURPOSE BATCH PLANTS[1, 2]

5.1 Introduction

The review in Chapter 2 highlights that a significant amount of work exists for scheduling MBPs. However, most models consider materials and equipment as the only resources. Furthermore, given the complex nature of operations in MBPs, these models usually simplify the problem by assuming one or more of a number of typical characteristics of MBPs. These assumptions include simultaneous and instantaneous material transfers, sequence-independent transition or setup times, no discrete resources (e.g. human), unlimited waste storage and treatment capacity, etc. Such assumptions hinder the application of such models to practical problems, as they do not assist in generating practically feasible schedules. Recently, Gimenez et al. [207, 208] presented a sequence of two papers for the short-term scheduling of the batch plants and include preventive maintenance, sequence-dependent cleaning times, non-zero material transfer times, and intermediate product delivery dates explicitly in their model. However, their models use a large number of binary variables and thus, require high computation time even for small examples. Later, the authors proposed strategies

---

[1] Susarla, N., Li, J., & Karimi, I. (2010). Resource constrained short-term scheduling for multipurpose batch plants. PSE Asia 2010, Singapore.
[2] Susarla, N., Li, J., & Karimi, I. A. (2009). Unit-slots based short-term scheduling for multipurpose batch plants. Presented in PSE 2009, Salvador-Bahia-Brazil.

[212] to improve the solution time of their models. Here, they introduced a set of additional slack variables to reduce the number of binary variables and make the formulation easier to solve. However, the inception of additional variables does not reduce the use of binary variable significantly and also, increases the model size. Clearly, further work is required to develop effective approaches that consider a number of practical features in scheduling MBPs.

Our specific goals in this chapter are (i) to extend the multi-grid model of Susarla et al. [206] to develop a continuous-time, multi-grid scheduling model considering resources other than materials and equipment; (ii) to modify, enhance, and extend the single-grid model of Sundaramoorthy & Karimi [70]. In addition, for both models, we consider a number of real-life scenarios such as the effect of various resources (such as utilities, human, waste treatment capacity, and material storage) on the production scheduling of the MBPs, sequence-dependent cleaning times, non-zero transfer times, and non-simultaneous material transfers. Also, our models allow variable batch sizes and processing times, multiple storage configurations (Classes: UIS, LIS, and FIS with policies: UW, LW, and NW), different scheduling objectives (such as profit maximization and makespan minimization), and a variety (limited and unlimited) of resources and utilities. We further give different variations of our models to appropriately suite their application to a given problem and exhibit better performance. We highlight different modelling limitations and requirements that significantly affect the performance with varying problem characteristics. We then present an extensive evaluation of examples from the literature and compare our results to the best known models. Through our extensive numerical evaluation, we further shed light on various strategies that affect solution time.

5.2 Problem Statement

An MBP produces a number of products using J common batch processing equipment or units, where $J = \{j = j1, j2, j3, ...\}$. The production operations involve S material states (including raw materials, intermediates, wastes, and products) and I unit operations or production steps or simply tasks, where $S = \{s = s1, s2, s3, ...\}$ and $I = \{i = i1, i2, i3, ...\}$. In addition, tasks in MBP require R resources other than materials and units such as human (operators) and utilities (e.g. steam, water, and solvents), where $R = \{r = r1, r2, r3, ...\}$. Such resources are either limited or unlimited by their total amount or rate of availability. This constrains plant operation and affects overall production cost. We describe the operation of MBP through the recipes [206] of various products, where a material state is any material (raw material, intermediate, waste, or product) with distinct attributes and properties. To describe the multipurpose and specialty nature of units, we define $I_j = \{i \mid \text{unit } j \text{ can process task } i\}$ and to highlight the resource intensive nature of tasks, we define $I_r = \{i \mid \text{resource } r \text{ is used by task } i\}$.

We consider that each batch of a task i in unit j has a batch size of b_{ij}. Then we assume that this batch requires a processing time of $\alpha_{ij} + \beta_{ij}b_{ij}$. Also, we assume that this batch of size b_{ij} requires $\mu_{ij} + \nu_{ij}b_{ij}$ amount (or rate) of resource r, where $i \in \mathbf{I}_j, \mathbf{I}_r$. Each such batch may consume a combination of multiple raw materials or intermediates and produce a combination of other materials (e.g. intermediates, wastes, and products). We define a mass ratio parameter, σ_{sij} [206], to quantify the actual (not net) amount of material s that a task i in unit j consumes or produces. This is defined as follows.

$$\sigma_{sij} = \pm \frac{\text{Mass of material state } s \text{ that task } i \text{ in unit } j \text{ consumes or produces}}{b_{ij}}$$

where, $\sigma_{sij} \begin{cases} = 0 & i \notin \mathrm{I}_j \text{ or task } i \text{ on unit } j \text{ does not involve } s \\ > 0 & i \in \mathrm{I}_j \text{ \& task } i \text{ on unit } j \text{ produces } s \\ < 0 & i \in \mathrm{I}_j \text{ \& task } i \text{ on unit } j \text{ consumes } s \end{cases}$

$$\sum_{s:\sigma_{sij}>0} \sigma_{sij} = \sum_{s:\sigma_{sij}<0} |\sigma_{sij}| = 1 \qquad i \in \mathbf{I}_j$$

We allow a known sequence-dependent transition time between the consecutive batches of two different tasks for cleaning and set-up of equipment. Each material state *s* (raw material, intermediate, waste, or product) has a dedicated storage, again denoted as s, and any of the three wait policies (unlimited-, limited-, or no-wait). The storage for each material state has a varied capacity. The storage capacities are either zero (no storage, NS), limited (limited storage, LS), or very high (unlimited storage, US). If a material does not have a storage space, we specify its storage capacity as zero. This allows us to demand that every unit consumes materials from their respective storage tanks and produce into respective storage tanks. In other words, we do not allow storage bypassing. For this, we assume that the difference between the material transfer times between two units by bypassing the storage and not bypassing it, are negligible. We further allow non-zero transfer times and non-simultaneous material transfers, into or out of a unit or storage. In addition, we allow the processing units to store materials if the storage or the downstream unit is not available.

Now, operations in MBPs are highly resource intensive. These resources can be broadly classified as resources limited by the rate of availability *RL* (e.g., utilities, human, etc.) and resources available in bulk quantities *BR*, which are usually procured as and when required (e.g., solvents, electricity, etc.). Resources limited by their rate of availability (*RL*) constrain plant operations by limiting the maximum use of a resource at any given time, where *RL* = {*r* is a resource limited by the rate of availability}. For example, in a plant with 2 operators, multiple tasks each requiring more than 1 operator cannot be performed simultaneously. Resources available in bulk quantities (*BR*) do not constrain plant operations but affect the cost of production, where *BR* = {*r* is a resource available in bulk quantities}.

With this, the MBP scheduling problem addressed in this article can be described as follows.

Given:

1. Recipes, materials, tasks, and mass ratios.
2. Processing units, their suitable tasks, and batch size limits
3. Initial material inventories, storage capacities, and wait policies
4. Resources, their availabilities, and requirements for each task
5. Sequence-dependent transition times and material transfer times
6. Market price of each material state

We determine:

1. The optimal sequence of the tasks and their schedules on each unit
2. Batch size of each task
3. Resource utilization and inventory profiles

For this, we assume:

1. Deterministic scenario i.e., no operational disruptions
2. Batch size dependent processing times and resource consumption
3. Material transfer times independent of distance between the units but dependent on size of the transfer duct and pumping power.

We consider two alternative objectives of revenue maximization for a given scheduling horizon [0, H] and makespan minimization for producing specified demands of products. Unless otherwise indicated, an index takes all its legitimate vales in all expressions and constraints in our formulation.

5.3 Multi-grid formulation using unit-slots

To schedule tasks on each unit *j*, material transfers into and out of storage *s*, and usage of a resource *r*, we model the horizon [0, H] in terms of K contiguous slots of unknown and arbitrary lengths, where **K** = (k = k1, k2, k3, ...). Figure 5.1 shows the schematic of our unit slots on a unit, a storage, and a resource. We consider the time before the beginning of the scheduling horizon is as slot zero (k0). Let, t_{jk} [$k \in \mathbf{K}$ + {k0}; $t_{j0} \geq 0$; $t_{jk} \leq$ H], t_{sk} [$k \in \mathbf{K}$ + {k0}; $t_{s0} \geq 0$; $t_{sk} \leq$ H], and t_{rk} [$k \in \mathbf{K}$ + {k0}] denote the end time of the slot *k* on processing unit *j*, storage *s*, and resource *r*, respectively. Thus, a slot *k* on unit *j*, storage *s*, and resource *r* has a length $t_{jk} - t_{j(k-1)}$, $t_{sk} - t_{s(k-1)}$, $t_{rk} - t_{r(k-1)}$.

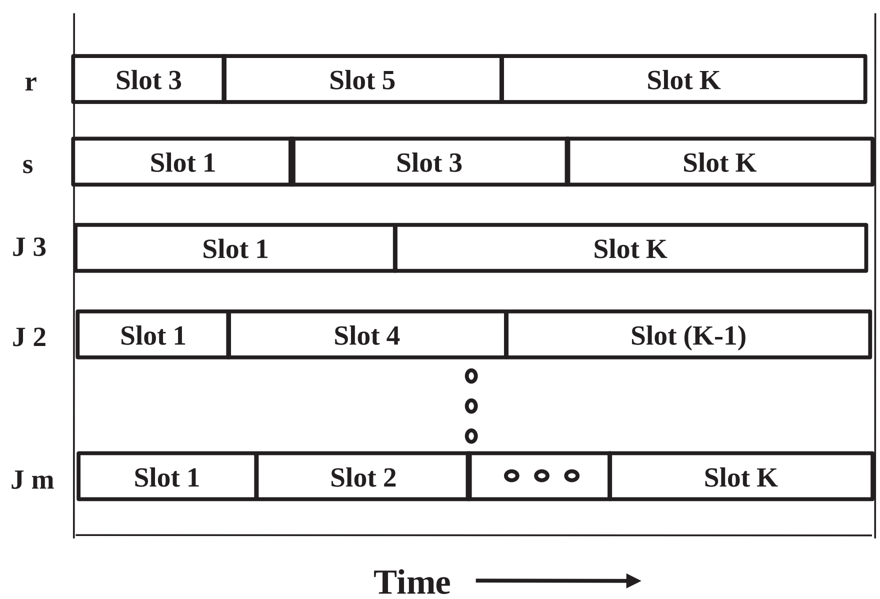

Figure 5.1 Design of unit-slots across units (J), storage (s), and resources (r)

Following Susarla et al. [206], we allow a task to start or end at any time during the scheduling horizon. Then, to allow sequence-dependent changeovers and non-simultaneous material transfers into and out of a unit *j*, we modify the construction of a unit slot appropriately. Figure 5.2 shows the modified construction of a unit-slot adopted in our formulation. By definition,

$$t_{j(k+1)} \geq t_{jk} \qquad k \in \mathbf{K} + \{k0\}, j \in \mathbf{J} \tag{5.1a}$$

$$t_{s(k+1)} \geq t_{sk} \qquad k \in \mathbf{K} + \{k0\}, s \in \mathbf{S} \tag{5.1b}$$

$$t_{r(k+1)} \geq t_{rk} \qquad k \in \mathbf{K} + \{k0\}, r \in \mathbf{R} \tag{5.1c}$$

We allocate tasks to slots on units such that every slot on a unit has one batch of a task and not more than one task is allocated to any slot. A slot k, $[t_{j(k-1)}, t_{jk}]$, on unit j processing a batch of any task involves four operations in the following order [206], as also shown in Figure 5.2.

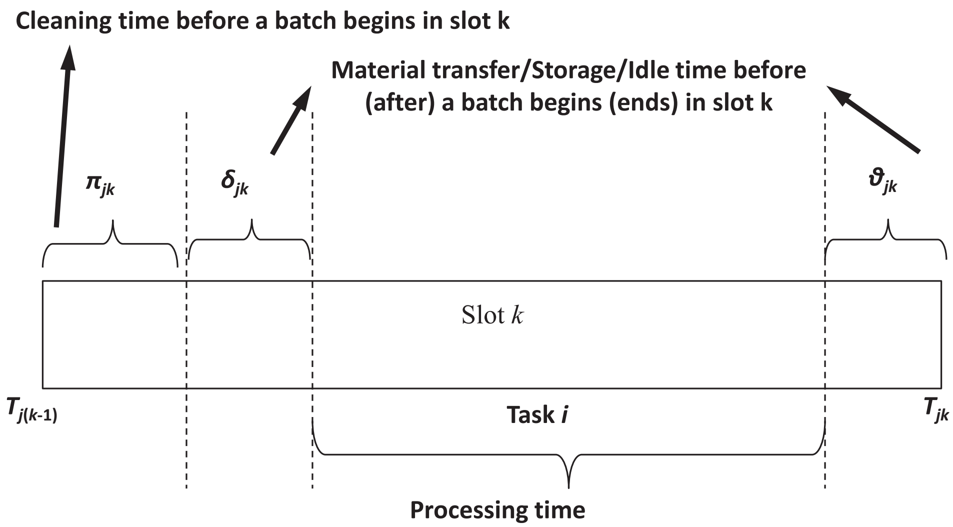

Figure 5.2 Novel construction of a unit-slot with four operations

1. Unit j is cleaned and prepared for the task during $[t_{j(k-1)}, t_{j(k-1)}+\pi_{jk}]$ allowing an appropriate time based on the task in the slot (k–1), where $\pi_{jk} \geq 0$ ($k \in \mathbf{K}$) is an unknown continuous variable and denotes the sequence-dependent changeover time.

2. The unit idles or receives the necessary input (raw or intermediate) materials for the current task during $[t_{j(k-1)}+\pi_{jk},\ t_{j(k-1)}+\pi_{jk}+\delta_{jk}]$, where $\delta_{jk} \geq 0$ ($k \in \mathbf{K}$) is an unknown continuous variable. δ_{jk} also denotes the delay in the actual start of a task after the changeover time during slot k.

3. The unit begins processing the batch at $t_{j(k-1)}+\pi_{jk}+\delta_{jk}$ and ends at $t_{jk} - \theta_{jk}$, where $\theta_{jk} \geq 0$ ($k \in \mathbf{K}$) is again an unknown continuous variable. θ_{jk} denotes the early end of the current task in slot k of unit j.

4. Then, the unit either idles or discharges the processed materials (output materials, e.g. intermediates, products, etc.) during $[t_{jk}-\theta_{jk}, t_{jk}]$. All the necessary transfers end by t_{jk}.

The material transfers into and out of processing units are neither instantaneous nor simultaneous. In other words, we consider material transfer times and allow non-simultaneous transfers of materials into or out of units or storages in our formulation. Now, a task may not begin and end within a slot, i.e. a task may require more than one slot to complete all of its operations (changeover, processing, and transfers). So, a unit does not stay idle (i.e. $\theta_{jk} = 0$) in slot k, if the unfinished task does not end and continues into the next slot. Similarly, a unit does not idle (i.e. $\delta_{jk} = 0$) and does not require any cleaning time (i.e. $\pi_{jk} = 0$) during slot k, if the current task does not begin in slot k but an unfinished task continues from the previous slot (k–1).

Following Susarla et al. [206], we define one binary (ys_{ijk}) and two 0-1 continuous (yr_{ijk} and ye_{ijk}) variables to respectively identify the beginning, continuation, and end of a task, which includes an idle task ($i0$) representing idling of a unit, as follows.

$$ys_{ijk} = \begin{cases} 1 & \text{if an allocation of task } i \text{ begins in slot } (k+1) \text{ of unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$i \in \mathbf{I}_j + \{i0\},\ k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K}$$

$$yr_{ijk} = \begin{cases} 1 & \text{if an allocation of task } i \text{ continues from slot } k \text{ to } k+1 \text{ in unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$i \in \mathbf{I}_j + \{i0\},\ k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K}$$

$$ye_{ijk} = \begin{cases} 1 & \text{if an allocation of task } i \text{ ends in slot } k \text{ of unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$i \in \mathbf{I}_j + \{i0\},\ k \in \mathbf{K}$$

By definition, ys_{ijk} denotes the beginning of an allocation for task i in unit j at time t_{jk}, yr_{ijk} represents the continuation of an unfinished run of task at t_{jk}, which continues from slot k to k+1, and ye_{ijk} denotes the end of an allocation of task i in unit j at t_{jk}. We assume that all batches that began before or during the scheduling horizon under consideration must end within the horizon. In other words, we do not allow any unfinished tasks at the end of H. Thus, we do not define the variables ys_{ijk} and yr_{ijk} at the end of H. Similarly, we fix the variables $ye_{ijk0} = 1$, if a batch of task i ends in slot 0 of unit j and $ye_{ijk0} = 0$ for all other tasks.

During any slot, each unit j must either be idle, begin a new task allocation or continue the previous allocation. So, we write,

$$\sum_{i \in \mathbf{I}_j + \{i0\}} (ys_{ijk} + yr_{ijk}) = 1 \qquad k \in \mathbf{K} + \{k0\} \tag{5.2}$$

The current allocation of a task ends only if the task begins a new allocation or continues an unfinished allocation in slot k and does not continue its run into next slot (k+1). This gives the following.

$$ye_{ijk} = yr_{ij(k-1)} + ys_{ij(k-1)} - yr_{ijk} \qquad i \in \mathbf{I}_j + \{i0\},\ k \in \mathbf{K},\ k \neq \mathbf{K} \tag{5.3a}$$

$$ye_{ijk} = yr_{ij(k-1)} + ys_{ij(k-1)} \qquad i \in \mathbf{I}_j + \{i0\},\ k \in \mathbf{K},\ k = \mathbf{K} \tag{5.3b}$$

Next, we define $b_{ijk} = B_{ij}^{L} ys_{ijk} + \Delta b_{ijk}$ ($i \in \mathbf{I}_j$, $k \in \mathbf{K}$) as the total amount of input materials for a batch of task i entering in slot (k + 1) of unit j, where B_{ij}^{L} is the minimum necessary amount of materials for task i in unit j and Δb_{ijk} is differential amount of materials over and above B_{ij}^{L} entering the batch in slot (k + 1). Let br_{ijk} denote the size of a batch of task that runs through slot k and continues in (k + 1) and similarly, be_{ijk} denote the total amount of output materials of task i that leave unit j by the end of slot k. We set br_{ijk0} = 0, if unit j is empty during slot 0 or does not continue any task into slot 1 else, we set an appropriate value to br_{ijk0}. As we demand all tasks beginning before or during H to end within H, we set br_{ijK} = 0. With this, we write the following balance on the amount of materials processed by a batch of task i in unit j at time t_{jk}.

$$be_{ijk} = br_{ij(k-1)} + B_{ij}^{L} ys_{ij(k-1)} + \Delta b_{ij(k-1)} - br_{ijk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K} \tag{5.4}$$

To ensure that the size of a batch entering, continuing, or ending in any unit do not exceed maximum allowable batch size B_{ij}^{U} at any point of time, we write the following bounds.

$$\Delta b_{ijk} \leq (B_{ij}^{U} - B_{ij}^{L}) ys_{ijk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K} \tag{5.5a}$$

$$br_{ijk} \leq B_{ij}^{U} yr_{ijk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K} + \{k0\} \tag{5.5b}$$

$$be_{ijk} \leq B_{ij}^{U} ye_{ijk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K} \tag{5.5c}$$

As described earlier, π_{jk} = 0, δ_{jk} = 0 and $\theta_{j(k-1)}$ = 0, whenever the allocation of a task does not begin in slot k but continues from slot (k – 1) and does not end but continues into slot (k + 1). So, we write the following.

$$\delta_{jk} = H \sum_{i \in \mathbf{I}_j} ys_{ijk} \qquad k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K} \tag{5.6a}$$

$$\pi_{jk} = H \sum_{i \in \mathbf{I}_j} ys_{ijk} \qquad k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K} \tag{5.6b}$$

$$\theta_{jk} = H \sum_{i \in \mathbf{I}_j} ye_{ijk} \qquad k \in \mathbf{K} \tag{5.6c}$$

We define tp_{jk} as the remaining processing time of an unfinished batch at the end of slot k. So, tp_{jk} must be zero if a task ends in slot k and does not continue into slot (k + 1). For this, we write the following.

$$tp_{jk} \leq \sum_{i \in \mathbf{I}_j} (\alpha_{ij} yr_{ijk} + \beta_{ij} br_{ijk}) \qquad k \in \mathbf{K}, o(k) \neq \mathrm{n}(\mathbf{K}) \tag{5.7}$$

Now, to ensure an appropriate changeover time between two different tasks, we demand that π_{jk} is greater than the sequence-dependent changeover time ($\tau_{ii'}$). For this, we must identify the last task in a unit. Thus, we define a 0-1 continuous variable x_{ijk} as follows.

$$x_{ijk} = \begin{cases} 1 & \text{if } i \text{ is the latest/current allocation in the slot } k \text{ of unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$i \in \mathbf{I}_j, k \in \mathbf{K} + \{k0\}, k \neq \mathbf{K}$$

We set $x_{ijk0} = 1$, if unit j continues an unfinished task i of slot 0 in slot 1 else, we set $x_{ijk0} = 0$. Clearly, only one task can be the latest allocation at any time T_{jk}. Then, if unit j processes a non-idle task i (begins/continues/ends) during slot k, then i must be the latest allocation. Thus, we write the following.

$$\sum_{i \in \mathbf{I}_j} x_{ijk} = 1 \qquad k \in \mathbf{K} + \{k0\}, k \neq \mathbf{K} \tag{5.8}$$

$$x_{ijk} \geq ys_{ijk} \qquad i \in \mathbf{I}_j, k \in \mathbf{K} + \{k0\}, k \neq \mathbf{K} \tag{5.9}$$

On the other hand, if unit j is idle during slot k, then the latest task would be the one in the previous slot. Also, the latest allocation in slot $(k + 1)$ must be same as the task in the slot k, if the task in slot k does not end or an idle task begins in slot $(k + 1)$. x_{ijk} allows us to identify the previous non-idle task on a unit j, hence facilitate modelling of sequence-dependant changeover time. Thus, we have,

$$x_{ijk} \geq x_{ij(k-1)} + ys_{(i0)jk} - 1 \qquad i \in \mathbf{I}_j, k \in \mathbf{K}, k \neq \mathbf{K} \tag{5.10}$$

$$x_{ijk} \geq x_{ij(k-1)} - \sum_{i' \in \mathbf{I}_j + \{i0\}} ye_{i'jk} \qquad i \in \mathbf{I}_j, k \in \mathbf{K}, k \neq \mathbf{K} \tag{5.11}$$

We demand that π_{jk} must be greater than the sequence-dependent changeover time ($\tau_{ii'}$). Thus, we write the following.

$$\pi_{j(k+1)} \geq \tau_{ii'}(ys_{ijk} + x_{i'j(k-1)} - 1) \qquad i, i' \in \mathbf{I}_j, k \in \mathbf{K}, k \neq \mathbf{K}, \tau_{ii'} > 0 \tag{5.12}$$

Clearly, the actual time of processing a batch in a given slot is calculated as tp_{jk} + $\sum_{i \in \mathbf{I}_j}[(\alpha_{ij} + \beta_{ij} B_{ij}^L) ys_{ijk} + \beta_{ij} \Delta b_{ijk}]$ – $tp_{j(k+1)}$. With this, we can modify Eqn. (5.1a) as following.

$$t_{j(k+1)} = t_{jk} + \pi_{j(k+1)} + \delta_{j(k+1)} + \sum_{i \in \mathbf{I}_j}[(\alpha_{ij} + \beta_{ij} B_{ij}^L) ys_{ijk} + \beta_{ij} \Delta b_{ijk}] + \theta_{j(k+1)} + tp_{jk} - tp_{j(k+1)}$$

$$k \in \mathbf{K}, k \neq \mathbf{K} \tag{5.13}$$

For each batch in a unit, a number of input materials (and resources) are transferred from respective storages to the unit, before the beginning of the batch. The processed materials (and used/spent resources) are then transferred from the unit to their respective storages. γ_{sj} gives the average transfer time for a unit mass of material *s* from its storage to unit *j* or vice-versa. The total transfer time required to transfer a material *s* from its storage to a unit *j* is given by $\gamma_{sj}\sigma_{sij}(B_{ij}^L ys_{ijk} + \Delta b_{ijk})$, where $\sigma_{sij} < 0$. Similarly, the time required to transfer a material *s* from a unit *j* to its storage is calculated as $\gamma_{sj}\sigma_{sij} be_{ijk}$, where $\sigma_{sij} > 0$. This inward and outward transfer of materials from storages is registered on the slots of respective storage. For this, we demand the transfer of a material from its storage must begin at the beginning of a slot on the storage and must end before the end of a slot.

In a multi-grid formulation, the timings and slots are not synchronized across storages, resources, and units. So, to ensure correct material (or resource) balances, it is really important to appropriately synchronize times for all such transfers. For this, we generalize the unit-slots approach of Susarla et al. [206] to demand that all transfers between any two entities (e.g. unit–unit, unit–storage, and unit–resource) must happen in the same slot. Consider the beginning of an allocation of task *i* in slot (k + 1) of unit *j*. Then, all input materials (or resources) required for this batch of task *i* must be transferred in the slot $k + 1$ of respective storages (or resources). Unit j must receive all input materials in the slot ($k + 1$) of unit j. This transfer begins at t_{sk} and is of duration $\gamma_{sj}\sigma_{sij}(B_{ij}^{L} ys_{ijk} + \Delta b_{ijk})$, where $\sigma_{sij} < 0$. As described earlier, these materials are received in unit j during $[t_{jk}+\pi_{j(k+1)}, t_{jk}+\pi_{j(k+1)}+\delta_{j(k+1)}]$. Thus, we write the following.

$$t_{sk} \geq t_{jk} + \pi_{j(k+1)} - H(1 - \sum_{i \in \mathbf{I}_j, \sigma_{sij}<0} ys_{ijk})$$

$$s: \sum_{i \in \mathbf{I}_j, \sigma_{sij}<0} \sigma_{sij} < 0,\ k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K} \tag{5.14a}$$

$$t_{sk} - \gamma_{sj}\sigma_{sij}(B_{ij}^{L} ys_{ijk} + \Delta b_{ijk}) \leq t_{jk} + \pi_{j(k+1)} + \delta_{j(k+1)} + H(1 - \sum_{i \in \mathbf{I}_j, \sigma_{sij}<0} ys_{ijk})$$

$$s: \sum_{i \in \mathbf{I}_j, \sigma_{sij}<0} \sigma_{sij} < 0,\ k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K} \tag{5.14b}$$

Also, unit j must receive necessary resources in slot ($k + 1$). The transfer of resource r begins at t_{rk}. We assume that resources are readily available and so, are instantaneously transferred to the unit. However, we can easily relax such an assumption using appropriate transfer times for resources, as we did for input materials in Eqn. (14). Since, a resource r is received in unit j during $[t_{jk}+\pi_{j(k+1)}, t_{jk}+\pi_{j(k+1)}+\delta_{j(k+1)}]$, we write,

$$t_{rk} \geq t_{jk} + \pi_{j(k+1)} - H(1 - \sum_{i \in \mathbf{I}_r} ys_{ijk})$$

$$r \in \mathbf{R},\ k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K} \tag{5.15a}$$

$$t_{rk} \leq t_{jk} + \pi_{j(k+1)} + \delta_{j(k+1)} + H(1 - \sum_{i \in \mathbf{I}_r} ys_{ijk})$$

$$r \in \mathbf{R},\ k \in \mathbf{K} + \{k0\},\ k \neq \mathbf{K} \tag{5.15b}$$

We now use a similar approach to address the transfer of materials and resources from a unit to storage. Consider the end of a task i in slot k of unit j. Unit j must transfer out all materials in the slot k. Also, the storage must receive such a material or resource in the same slot k. We demand that the transfer of output materials end at t_{sk}

and of resources end at t_{rk}. We assume instantaneous transfer of resources but the transfer of materials require a duration of $\gamma_{sj}\sigma_{sij}be_{ijk}$, where $\sigma_{sij} > 0$. Now, this transfer of materials and resources must occur during $[t_{jk}–\theta_{jk},\ t_{jk}]$. Therefore, we write the following Eqn. (16) for the transfer of materials and Eqn. (17) for the transfer of resources.

$$t_{sk} - \gamma_{sj}\sigma_{sij}be_{ijk} \geq t_{jk} - \theta_{jk} - H(1 - \sum_{i\in \mathbf{I}_j, \sigma_{sij}>0} ye_{ijk}) \quad s\text{:} \sum_{i\in \mathbf{I}_j, \sigma_{sij}<0} \sigma_{sij} < 0,\ k \in \mathbf{K} \qquad (5.16a)$$

$$t_{sk} \leq t_{jk} + H(1 - \sum_{i\in \mathbf{I}_j, \sigma_{sij}>0} ye_{ijk}) \qquad s\text{:} \sum_{i\in \mathbf{I}_j, \sigma_{sij}>0} \sigma_{sij} > 0,\ k \in \mathbf{K} \qquad (5.16b)$$

$$t_{rk} \geq t_{jk} - \theta_{jk} - H(1 - \sum_{i\in \mathbf{I}_r \cap \mathbf{I}_j} ye_{ijk}) \qquad r \in \mathbf{R},\ k \in \mathbf{K} \qquad (5.17a)$$

$$t_{rk} \leq t_{jk} + H(1 - \sum_{i\in \mathbf{I}_r \cap \mathbf{I}_j} ye_{ijk}) \qquad r \in \mathbf{R},\ k \in \mathbf{K} \qquad (5.17b)$$

Aforementioned synchronization of material transfers (Eqn. 5.14–5.17) across different time-grids of storages and units allows us to write the following balance on the quantity of each material in its storage at the end of each slot.

$$q_{sk} = q0_s + \sum_{j\in \mathbf{J}} \sum_{i\in \mathbf{I}_j, \sigma_{sij}<0} \sigma_{sij}(B_{ij}^L ys_{ijk} + \Delta b_{ijk}) \qquad k \in \{k0\} \qquad (5.18a)$$

$$q_{sk} = q_{s(k-1)} + \sum_{j\in \mathbf{J}} \sum_{i\in \mathbf{I}_j, \sigma_{sij}<0} \sigma_{sij}(B_{ij}^L ys_{ijk} + \Delta b_{ijk}) + \sum_{j\in \mathbf{J}} \sum_{i\in \mathbf{I}_j, \sigma_{sij}>0} \sigma_{sij}be_{ijk}$$

$$k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.18b)$$

$$q_{sk} = q_{s(k-1)} + \sum_{j\in \mathbf{J}} \sum_{i\in \mathbf{I}_j, \sigma_{sij}>0} \sigma_{sij}be_{ijk} \qquad k \in \mathbf{K},\ k = \mathbf{K} \qquad (5.18c)$$

where, q_{sk} ($Q_s^L \leq q_{sk} \leq Q_s^U$) is the net quantity of material s present in its storage at the end of slot k.

We assume that for non-discrete (uncountable) resources, such as utilities, the amount of resource consumption is directly proportional to the batch size. Also, we consider that for discrete (countable) resources, such as human, it directly depends on the task itself. μ_{ri} gives information about the consumption of each resource by a task i as follows.

1. For uncountable $r \in$ **RL**, μ_{ri} is the rate of resource r required per unit batch size.

2. For countable $r \in$ **RL**, μ_{ri} is the number of discrete resource r required for task i.

3. For $r \in$ **BR**, μ_{ri} is the amount of resource r required per unit batch size.

With this, we write the following balance on the usage of non-discrete resource $r \in$ **RL**.

$$a_{rk} = \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \lambda_{ri} (B_{ij}^L ys_{ijk} + \Delta b_{ijk}) \qquad r \in \mathbf{RL},\ k \in \{k0\} \tag{5.19a}$$

$$a_{rk} = a_{r(k-1)} + \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \lambda_{ri} (B_{ij}^L ys_{ijk} + \Delta b_{ijk}) - \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \lambda_{ri} be_{ijk}$$

$$r \in \mathbf{RL},\ k \in \mathbf{K},\ k \neq \mathbf{K} \tag{5.19b}$$

$$a_{rk} = a_{r(k-1)} - \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \lambda_{ri} be_{ijk} \qquad r \in \mathbf{RL},\ k \in \mathbf{K},\ k = \mathbf{K} \tag{5.19c}$$

where, a_{rk} ($a_{rk} \leq A_r^U$) denotes the total rate of resource r being consumed at t_{jk}. A_r^U gives the maximum rate of resource available.

For discrete resource $r \in$ **RL**, we modify Eqn. 5.19 as the following. Here, a_{rk} denotes the total number of a resource r in use at t_{jk}.

$$a_{rk} = \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \mu_{ri} ys_{ijk} \qquad r \in \mathbf{RL},\ k \in \{k0\} \tag{5.20a}$$

$$a_{rk} = a_{r(k-1)} + \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \mu_{ri} ys_{ijk} - \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \mu_{ri} ye_{ijk} \qquad r \in \mathbf{RL},\ k \in \mathbf{K},\ k \neq \mathbf{K} \tag{5.20b}$$

$$a_{rk} = a_{r(k-1)} - \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \mu_{ri} ye_{ijk} \qquad r \in \mathbf{RL},\ k \in \mathbf{K},\ k = \mathbf{K} \tag{5.20c}$$

Then, for both discrete and non-discrete resource $r \in$ **BR**, we monitor the overall consumption with the following, respectively.

$$a_{rk} = a_{r(k-1)} + \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \mu_{ri} ys_{ijk} \qquad r \in \mathbf{RL}, k \in \mathbf{K} \tag{5.21}$$

$$a_{rk} = a_{r(k-1)} + \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \mu_{ri} (B_{ij}^{L} ys_{ijk} + \Delta b_{ijk}) \qquad r \in \mathbf{RL}, k \in \mathbf{K} \tag{5.22}$$

We present the two most used scheduling objectives in the literature: maximization of revenue and minimization of makespan. υ_s gives the price of a unit quantity of material *s*. So, the revenue *rev* from production is given by:

$$rev = \sum_{k \in \mathbf{K}} \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j} \sum_{s:\sigma_{sij} > 0} \sigma_{sij} \upsilon_s be_{ijk} \tag{5.23}$$

For the second objective, i.e. makespan minimization, H ceases to be a given parameter. Instead, we need to satisfy a given demand (D_s) for each material *s*. So, we write the following.

$$D_s \le \sum_{k \in \mathbf{K}} \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j} \sum_{s:\sigma_{sij} > 0} \sigma_{sij} be_{ijk} \tag{5.24}$$

Here, we use all of the following to compute makespan (*ms*), even though only the first would be sufficient.

$$ms \ge t_{jk} \qquad k \in \mathbf{K}, k = \mathbf{K} \tag{5.25a}$$

$$ms \ge t_{sk} \qquad k \in \mathbf{K}, k = \mathbf{K} \tag{5.25b}$$

$$ms \ge t_{rk} \qquad k \in \mathbf{K}, k = \mathbf{K} \tag{5.25c}$$

This completes our multi-grid model (Eqn. 5.1 – Eqn. 5.25) for scheduling resource-constrained MBPs with a variety of resource constraints, non-simultaneous material transfers, non-zero material transfer times, and sequence dependent changeover times.

5.4 Single-grid formulation using process-slots

Here, we develop a single-grid formulation for scheduling MBPs using process-slots. For this, we modify the basic model of Sundaramoorthy and Karimi [70] to construct an enhanced and extended formulation for scheduling MBPs. Our new formulation allows non-simultaneous material transfers, non-zero transfer times, sequence-dependent changeovers, and resources other than materials and equipment such as utilities and manpower.

Similar to the unit-slots approach, we divide the scheduling horizon [0, H] in **K** contiguous slots of unknown and variable lengths to schedule tasks on each unit *j*, to ensure correct material transfers into and out of storage *s*, and to monitor usage of a resource *r*, where $\mathbf{K} = \{k0, k1, k2, k3, ...\}$. Again, we consider the time before the beginning of the scheduling horizon is as slot zero (*k*0). However, unlike in unit slots approach, in process slots approach the slots are synchronized across all units, storages, and resources. A slot *k* begins at time $T_{(k\text{-}1)}$ and ends at time T_k and has a length of sl_k ($= T_k - T_{(k\text{-}1)}$), $k \in$ K. As the total length of all such process slots must be at most equal to the duration of the scheduling horizon, we write the following.

$$\sum_{k \in \mathbf{K}} sl_k \le H \tag{5.26}$$

Similar to unit slots, we allocate processing tasks to the process slots such that every slot has a task and each slot has only one task. As stated earlier, each task involves four operations namely, cleaning, inward material transfer, processing, and outward material transfer. Let $o \in \mathbf{O} = \{o1, o2, o3, o4\}$ denote such an operation, where *o*1, *o*2, *o*3 and *o*4 represents the operations of cleaning, inward material transfer, task processing, and outward material transfer, respectively.

To this end, we define the following one binary and two 0-1 continuous variables to identify the start, continuation, and end of an operation of a task on any unit.

$$ys_{iojk} = \begin{cases} 1 & \text{if an allocation of operation } o \text{ of } i \text{ begins in slot } (k+1) \text{ of } j \\ 0 & \text{otherwise} \end{cases}$$

$$o \in \mathbf{O},\ i \in \mathbf{I}_j,\ k \in \mathbf{K},\ k \neq \mathbf{K}$$

$$yr_{iojk} = \begin{cases} 1 & \text{if an allocation of } o \text{ of } i \text{ continues from } k \text{ to } k+1 \text{ in unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$o \in \mathbf{O},\ i \in \mathbf{I}_j,\ k \in \mathbf{K},\ k \neq \mathbf{K}$$

$$ye_{iojk} = \begin{cases} 1 & \text{if an allocation of o of a task } i \text{ ends in slot } k \text{ of unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$o \in \mathbf{O},\ i \in \mathbf{I}_j,\ k \in \mathbf{K}$$

By definition, ys_{iojk} denotes the beginning of an allocation for perform operation o of a task i in unit j at time t_{jk}, yr_{ijk} represents the continuation of an unfinished run of the operation o of task at t_{jk}, which continues from slot k to k+1, and ye_{ijk} denotes the end of an allocation for operation o of task i in unit j at t_{jk}. We assume that all batches that began before or during the scheduling horizon under consideration must end within the horizon. In other words, we do not allow a new task to begin and any unfinished tasks to end at or after H. Thus, we do not define the variables ys_{iojk} and yr_{iojk} at the end of H. Similarly, we fix the variables $ye_{iojk0} = 1$, if a batch of task i ends its current operation o in slot 0 of unit j and $ye_{iojk0} = 0$ for all other tasks. Note that as we allow unit to idle inherently during material transfer and other operations, we do not have explicit idle tasks in our formulation. This along with explicit treatment of multiple operations of a processing task constitute a significant departure from the model of Sundaramoorthy and Karimi [70].

During any slot, each unit j may either begin or continue the allocation of a processing task i and perform only one of the four process operations o of that task. So, we write the following.

$$\sum_{o\in\mathbf{O}}\sum_{i\in\mathbf{I}_j}(ys_{iojk}+yr_{iojk})=1 \qquad k\in\mathbf{K},\ k\neq\mathbf{K} \tag{5.27}$$

At the end of any given slot, a task may either end its allocation (or end one of its operations) and begin a new allocation (or the next operation) at the start of the next slot or continue its current allocation (or the current operation) into the next slot. Also, as the sequence of the four operations constructing a task are known and fixed, we demand that the end of an operation represents the start of the next operation. We ensure all this in our formulation by the following.

$$ye_{iojk}=yr_{ioj(k-1)}+ys_{ioj(k-1)}-yr_{iojk} \qquad o\in\mathbf{O},\ i\in\mathbf{I}_j,\ k\in\mathbf{K},\ k\neq\mathbf{K} \tag{5.28}$$

$$ye_{iojK}=yr_{ioj(K-1)}+ys_{ioj(K-1)} \qquad o\in\mathbf{O},\ i\in\mathbf{I}_j \tag{5.29}$$

$$\sum_{i\in\mathbf{I}_j}ye_{iojk}=\sum_{i\in\mathbf{I}_j}ys_{i(o+1)jk} \qquad o\in\mathbf{O},\ o\neq\mathbf{O},\ k\in\mathbf{K},\ k\neq\mathbf{K} \tag{5.30a}$$

$$\sum_{i\in\mathbf{I}_j}ye_{io4jk}=\sum_{i'\in\mathbf{I}_j,i\neq i}ys_{i'o1jk} \qquad k\in\mathbf{K},\ k\neq\mathbf{K} \tag{5.30b}$$

Following our multi-grid formulation, we use $b_{ijk}=B_{ij}^{L}ys_{io3jk}+\Delta b_{ijk}$ ($i\in\mathbf{I}_j$, $k\in\mathbf{K}$, $k\neq\mathbf{K}$), br_{ijk} ($i\in\mathbf{I}_j$, $k\in\mathbf{K}$, $k\neq\mathbf{K}$), and be_{ijk} ($i\in\mathbf{I}_j$, $k\in\mathbf{K}$, $k\neq k0$) to denote the total amount of input materials for a batch of task i entering in slot (k + 1) of unit j, the size of a batch that runs through slot k and continues in (k + 1), and the total amount of output materials of task i that leave unit j by the end of slot k, respectively. B_{ij}^{L} is the minimum necessary amount of materials for task i in unit j and Δb_{ijk} is differential amount of materials over and above B_{ij}^{L} entering the batch in slot (k + 1). br_{ijk0} = 0, if unit j is empty during slot 0 or does not continue any task into slot 1 else, we set an appropriate value to br_{ijk0}. As we demand all tasks beginning before or during H to end within H, we set $br_{ijK} = 0$. The batch sizes (entering, continuing, or leaving a unit) are constrained by the maximum allowable size of a batch in the equipment B_{ij}^{U} . Therefore, we have the following bounds.

$$\Delta b_{ijk} \leq (B_{ij}^{U} - B_{ij}^{L}) ys_{io3jk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K},\ k \neq \mathbf{K} \tag{5.31a}$$

$$br_{ijk} \leq B_{ij}^{U} yr_{io3jk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K},\ k \neq \mathbf{K} \tag{5.31b}$$

$$be_{ijk} \leq B_{ij}^{U} ye_{io3jk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K},\ k \neq k0 \tag{5.31c}$$

Note that Eqs 5.31 are written only for the processing operation of a task ($o3$). We now write a balance on the amount of material processed by a batch of each task in a unit across a slot.

$$be_{ijk} = br_{ij(k-1)} + B_{ij}^{L} ys_{io3j(k-1)} + \Delta b_{ij(k-1)} - br_{ijk} \qquad i \in \mathbf{I}_j,\ k \in \mathbf{K},\ k \neq k0 \tag{5.32}$$

Now, every batch of a given task consumes a number of raw materials or intermediates and produces other intermediates or products. These materials are required to be transferred from the respective storage to the processing unit or vice-versa. Usually, this transfer of materials requires time. Also, a batch may consume (produce) multiple materials, which may not always be transferred to the processing units (storage) simultaneously. As discussed earlier, most models in the literature have ignored such transfer times. Thus, to account the issues of non-zero and non-simultaneous material transfer times, we define the following one binary and two 0-1 continuous variables.

$$yts_{sijk} = \begin{cases} 1 & \text{if the transfer of material } s \text{ for } i \text{ begins in slot } (k+1) \text{ of } j \\ 0 & \text{otherwise} \end{cases}$$

$$s \in \mathbf{S},\ i \in \mathbf{I}_j, \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq \mathbf{K}$$

$$ytr_{sijk} = \begin{cases} 1 & \text{if transfer of } s \text{ for } i \text{ continues from } k \text{ to } k+1 \text{ in unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq \mathbf{K}$$

$$yte_{sijk} = \begin{cases} 1 & \text{if transfer of } s \text{ for a task } i \text{ ends in slot } k \text{ of unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq k0$$

The transfer of materials occurs during the inward and outward transfer operations of a task. So, the material transfer decisions are synchronized with the start, continuation, and end of transfer operations. The transfer of a material may start, continue, or end at the end of any given slot only if a transfer operation is in progress at the same time in the unit. Thus, we have

$$\sum_{i \in \mathbf{I}_j} (yts_{sijk} + ytr_{sijk}) = \sum_{i \in \mathbf{I}_j} (ys_{io2jk} + yr_{io2jk} + ys_{io4jk} + yr_{io4jk}) \qquad k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.33)$$

$$yte_{sijk} = ytr_{sij(k-1)} + yts_{sij(k-1)} - ytr_{sijk} \qquad s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.34a)$$

$$yte_{sijK} = ytr_{sij(K-1)} + yts_{sij(K-1)} \qquad s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s \qquad (5.34b)$$

The raw materials required for a batch are transferred in to the unit before the beginning of the processing operation. For this, we demand that the transfer in of input materials may start at any time and at different times during the inward transfer operation ($o2$). However, we require that all such transfers must end together and at the same time as the end of the inward transfer operation. Thus, the end of inward transfer also signifies the start of the processing operation ($o3$). Similarly, we demand that the transfer out of all materials must start immediately with the outward transfer operation ($o4$) but end any time during the operation. Mathematically, we write all this is as the following.

$$yts_{sijk} \leq ys_{io2jk} + yr_{io2jk}$$

$$s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ s\colon \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} \sigma_{sij} < 0,\ \ k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.35a)$$

$$\sum_{s \in \mathbf{I}_s} yte_{sijk} = \left|\mathbf{I}_s\right| ye_{io2jk}$$

$$s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ s\!:\ \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} \sigma_{sij} < 0\,,\ \ k \in \mathbf{K},\ k \neq k0 \qquad (5.35b)$$

$$yte_{sijk} \leq ye_{io4jk} + yr_{io4jk}$$

$$s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ s\!:\ \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} \sigma_{sij} > 0\,,\ \ k \in \mathbf{K},\ k \neq k0 \qquad (5.36a)$$

$$\sum_{s \in \mathbf{I}_s} yts_{sijk} = \left|\mathbf{I}_s\right| ys_{io4jk}$$

$$s \in \mathbf{S},\ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ s\!:\ \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} \sigma_{sij} > 0\,,\ \ k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.36b)$$

Let as_{sijk}, ar_{sijk}, and ae_{sijk} represent the amount of material s transferred from storage (unit) to the processing unit (storage). The amount of material transferred in or out of a storage (processing unit) must be less than a maximum amount allowable by the transfer system. Also, we ensure that the total amount of material transferred in or out of a batch must be equal to the batch size. Therefore, we write the following along with a balance on the amount of material transferred.

$$ae_{sijk} \leq A_s^U yte_{sijk} \qquad i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq k0 \qquad (5.37a)$$

$$ar_{ijk} \leq A_s^U ytr_{sijk} \qquad i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.37b)$$

$$as_{ijk} \leq A_s^U yts_{sijk} \qquad i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.37c)$$

$$ae_{sijk} = ar_{sij(k-1)} + as_{sij(k-1)} - ar_{sijk} \qquad i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq k0 \qquad (5.38)$$

$$ae_{sijk} = -\sigma_{sij}\left(B_{ij}^L ys_{io3jk} + \Delta b_{ijk}\right)$$

$$s\!:\ \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} \sigma_{sij} < 0\,,\ \ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq \mathbf{K} \qquad (5.39a)$$

$$as_{sijk} = \sigma_{sij} be_{io3jk} \qquad s\!:\ \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} \sigma_{sij} > 0\,,\ \ i \in \mathbf{I}_j,\ \mathbf{I}_s,\ k \in \mathbf{K},\ k \neq k0 \qquad (5.39b)$$

where, A_s^U is the upper limit on the transfer of material s.

To allow an appropriate changeover time (π_{jk}) between two different tasks, we ensure that π_{jk} is greater than the sequence-dependent changeover time ($\tau_{ii'}$). Thus, we write the following.

$$\pi_{j(k+1)} \geq \tau_{ii'}(ys_{io1jk} + ye_{i'o4j(k-1)} - 1) \qquad i, i' \in \mathbf{I}_j, k \in \mathbf{K}, k \neq \mathbf{K} \tag{5.40}$$

The total length of an allocation for a task is the sum of times for each of the operations involved. $\pi_{j(k+1)} + \sum_{i \in \mathbf{I}_j, \mathbf{I}_s} \delta_{sij} as_{sijk} + \sum_{i \in \mathbf{I}_j} [(\alpha_{ij} + \beta_{ij} B_{ij}^L) ys_{io3jk} + \beta_{ij} \Delta b_{ijk}]$ gives the total time required for a task, where δ_{sij} is a parameter that represents the linear dependence of the time required to transfer as_{sijk} amount of material from its storage to unit j or vice-versa, α_{ij} and β_{ij} are the parameters that define the linear dependence of the batch processing time on the batch size. Thus, to monitor the amount of time left for a task to finish its allocation at a given time, we write the following balance on the amount of time left, t_{jk}.

$$t_{j(k+1)} \geq t_{jk} + \pi_{j(k+1)} + \sum_{i \in \mathbf{I}_j, \mathbf{I}_s} \delta_{sij} as_{sijk} + \sum_{i \in \mathbf{I}_j} [(\alpha_{ij} + \beta_{ij} B_{ij}^L) ys_{io3jk} + \beta_{ij} \Delta b_{ijk}] - sl_{(k+1)}$$

$$k \in \mathbf{K}, k \neq \mathbf{K} \tag{5.41}$$

Let q_{sk} ($Q_s^L \leq q_{sk} \leq Q_s^U$) denote the net quantity of material s available in its storage at the end of slot k. Next, we account for all material transfers in and out of a storage and write the following balance on the inventory of each material in its storage at the end of each slot.

$$q_{sk} = q0_s - \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} as_{sijk} \qquad k \in \{k0\} \tag{5.42a}$$

$$q_{sk} = q_{s(k-1)} - \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j, \sigma_{sij} < 0} as_{sijk} + \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} ae_{sijk}$$

$$k \in \mathbf{K}, k \neq \mathbf{K} \tag{5.42b}$$

$$q_{sk} = q_{s(k-1)} + \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j, \sigma_{sij} > 0} ae_{sijk} \qquad k \in \mathbf{K}, k = \mathbf{K} \tag{5.42c}$$

where, Q_s^L, Q_s^U are the minimum and maximum limits on the inventory of material s in its storage and $q0_s$ is the quantity of material *s* present at the start of the horizon.

Here, we have only discussed materials and equipment as our only resources. However, it is straight forward to account for other resources in our single-grid formulation. For this, we treat resources as other materials required by the processing tasks and then write a balance on their inventory.

We again use the two most common scheduling objectives, as also proposed in the unit-slots model: maximization of revenue and minimization of makespan. υ_s gives the price of a unit quantity of material *s*. So, the total revenue *rev* from the sales of products is given by:

$$rev = \sum_{k \in \mathbf{K}} \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j} \sum_{s:\sigma_{sij} > 0} \upsilon_s ae_{sijk} \tag{5.43}$$

For the second objective, i.e. makespan minimization, H ceases to be a given parameter. Instead, we need to satisfy a given demand (D_s) for each material *s*. So, we write the following.

$$D_s \le \sum_{k \in \mathbf{K}} \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j} \sum_{s:\sigma_{sij} > 0} ae_{ijk} \tag{5.44}$$

Next, to compute the makespan (*ms*) we modify the Eqn 26 and write the following.

$$\sum_{k \in \mathbf{K}} sl_k \le ms \tag{5.45}$$

With this, our single grid formulation for scheduling MBPs (SLK-sg) comprises eqs. 5.26 – 5.45.

5.5 Numerical Evaluation

To study the performance of our models, we consider three examples from the literature. We then evaluate the performance statistics against some of the best known models from the literature. A fair and an unbiased comparison demands careful attention on many factors [209] such as hardware, operating system, and software. In our study, we used CPLEX 11/GAMS 22.8 [210] on a Dell precision PWS690 workstation with Intel® Xeon® 3 GHz CPU, 16 GB RAM, running Windows XP Professional x64 Edition.

For the sake of a fair comparison with the literature models, we modify our single grid model to make it equivalent to others in terms of features and assumptions. We use three models from the literature for our study: MG [76], SF [82], and CBMN [75]. From now on, we will refer to these models as MG, SF, and CBMN. We refer our multi-grid models as SLK1 and SLK2 and our single-grid model as SLK-sg.

5.5.1 Example 1

This example is originally from Kondili et al. [34]. It illustrates the handling of utility constraints. Figure 5.3 gives the recipe diagram. It involves 3 processing units ($j1$ – $j3$), 4 tasks ($i1$ – $i4$), 7 materials ($s1$ – $s7$), and 2 utilities (cooling water and high pressure steam). $i1$ and $i2$ can be performed in either $j1$ or $j2$ whereas $i3$ and $i4$ can be run only on $j3$. While $i1$ and $i3$ require cooling water for their operation, $i2$ and $i4$ need high pressure steam. The relevant data for this example is given in Tables 5.1 and 5.2.

Table 5.1 Storage capacities, initial inventories, and material prices for Examples 1–3

	Example 1			Example 2			Example 3		
	Q_s^U	$q0_s$	υ_s	Q_s^U	$q0_s$	υ_s	Q_s^U	$q0_s$	υ_s
S1	1000	400	10	AA	100	0	100	100	0
S2	1000	400	15	AA	100	0	100	100	0
S3	200	0	25	0	0	0	5	0	0
S4	100	0	0	0	0	0	5	0	0
S5	500	0	0	15	0	0	3	0	0
S6	1000	0	30	40	10	0	3	0	0
S7	1000	0	40	0	0	0	100	0	1
S8				0	0	0	100	0	1
S9				UL	0	0			
S10				UL	0	0			
S11				UL	0	1			
S12				UL	0	1			
S13				UL	0	1			

Table 5.2 Batch size data for Example 1

	B_{ij}^{U}			B_{ij}^{L}			α_{ij}			β_{ij}			γ_{ij}			μ_{ri}		
	$j1$	$j2$	$j3$	$j1$	$j2$	$j3$	$j1$	$j2$	$j3$	$j1$	$j2$	$j3$	$j1$	$j2$	$j3$	$j1$	$j2$	$j3$
$i1$	80	50		40	25		0.5	0.5		0.025	0.4		6	4		0.25	0.25	
$i2$	80	50		40	25		0.75	0.75		0.0375	0.06		4	3		0.3	0.3	
$i3$			80			40			0.25			0.0125			8			0.4
$i4$			80			40			0.5			0.025			4			0.5

Table 5.3 Model and solution statistics for Example 1

Model	K	CPU time (s)	Nonzeros	Nodes	MILP	Binary variables	Continuous variables	Constraints
Example 1a: Revenue Maximization								
SLK1	**7**	**6.7**	**2640**	**3547**	**$5,904**	**54**	**401**	**684**
SLK2	**7**	**7.5**	**2628**	**4393**	**$5,904**	**54**	**371**	**666**
SLK-sg	**7**	**2.3**	**1501**	**1798**	**$5,904**	**54**	**314**	**284**
CBMN	**7***	**1.2**	**1288**	**1067**	**$5,904**	**66**	**158**	**334**
SF	**6***	**2.5**	**3218**	**1453**	**$5,904**	**66**	**200**	**850**
MG	**7**	**1.7**	**4117**	**975**	**$5,904**	**84**	**484**	**1052**
Example 1b: Revenue Maximization								
SLK1	**6**	**0.3**	**2197**	**240**	**$5,228**	**45**	**340**	**570**
SLK2	**6**	**0.4**	**2185**	**282**	**$5,228**	**45**	**313**	**555**
SLK-sg	**6**	**0.27**	**1201**	**197**	**$5,228**	**45**	**265**	**236**
CBMN	**6***	**0.17**	**1062**	**68**	**$5,228**	**54**	**133**	**280**
SF	**6***	**1.1**	**3218**	**1224**	**$5,228**	**66**	**200**	**850**
MG	**6**	**0.15**	**3356**	**67**	**$5,228**	**72**	**415**	**901**
Example 1c: Makespan Minimization								
SLK1	**8**	**4.7**	**3106**	**5362**	**8.5 h**	**63**	**466**	**811**
SLK2	**8**	**6**	**3094**	**5752**	**8.5 h**	**63**	**433**	**790**
SLK-sg	**8**	**3**	**1819**	**3128**	**8.5 h**	**63**	**363**	**333**
CBMN	**8***	**1.2**	**1515**	**1329**	**8.5 h**	**78**	**183**	**390**
SF	**7***	**0.87**	**3837**	**1200**	**8.5 h**	**78**	**233**	**1012**
MG	**8**	**4.3**	**5073**	**741**	**8.5 h**	**96**	**610**	**1212**
Example 1d: Makespan Minimization								
SLK1	**7**	**0.39**	**2663**	**318**	**9.025 h**	**54**	**405**	**697**
SLK2	**7**	**0.53**	**2651**	**434**	**9.025 h**	**54**	**375**	**676**
SLK-sg	**7**	**0.46**	**1501**	**362**	**9.025 h**	**54**	**314**	**285**
CBMN	**7***	**0.17**	**1289**	**81**	**9.025 h**	**66**	**158**	**336**
SF	**6***	**1.42**	**3229**	**44**	**9.025 h**	**66**	**200**	**857**
MG	**7**	**0.76**	**4241**	**190**	**9.025 h**	**84**	**534**	**1061**

For revenue maximization, we solve this example for two scenarios. In scenario 1a, we assume that the rate of availability for both utilities (cooling water and high pressure steam) is 30 Kg/min. Table 5.3 shows that, for this scenario, all models yield solutions with identical objective value of $ 5904 within a comparable solution time. However, our both single- and multi-grid formulations require fewer binary variables for this solution. In scenario 1b, we consider the utilities to be available at the rate of 40 Kg/min. Again, all models perform equally well and yield the same objective value of $ 5228 within comparable solution times. Also, the SLKs require fewer binary variables as compared to the other models.

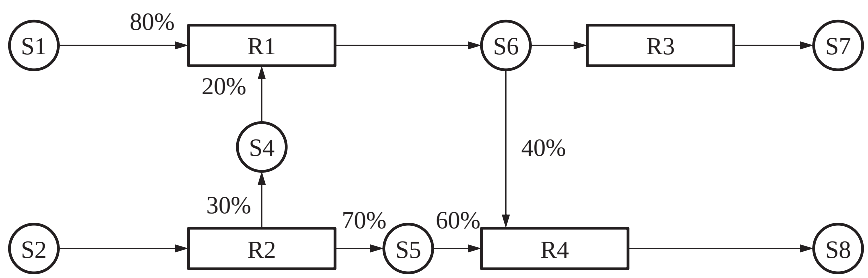

Figure 5.3 Recipe diagram for Example 1

For makespan minimization, we again solve this example for the two scenarios of 30 Kg/min and 40 Kg/min resource availability. The minimum makespan for these two scenarios are 8.5 h and 9.025 h. Table 5.3 lists the complete model and solution statistics. SLKs consistently require fewer binary variables for both scenarios for the same objective value. For both the scenarios, all models could obtain an optimal solution within a few CPU seconds.

5.5.2 Example 2

This example is from Maravelias and Grossmann [76]. It specifically demonstrates the capability of our models in handling zero wait policy for some of the intermediate material states and various storage capacities for others. Figure 5.4 gives the recipe diagram. Tables 5.1 and 5.4 list the data. It involves 6 processing units (*j*1 – *j*6), 10 tasks (*i*1 – *i*10), 14 materials (*s*1 – *s*14), and 3 utilities (cooling water, low pressure steam, and high pressure steam). Here, *j*1 is suitable for tasks *i*1 and *i*4, *j*2 for *i*2, *j*3 for *i*3, *j*4 for *i*5 and *i*6, *j*5 for *i*7 and *i*9, and *j*6 for *i*8 and *i*10. We consider that an unlimited storage is available for raw materials *s*1 and *s*2, intermediates *s*9 and *s*10, and final products *s*11-*s*13; limited storage is available for materials *s*5 and *s*6; no intermediate storage is available for states *s*4 and *s*8; and a zero wait policy applies for materials *s*3 and *s*7. Tasks *i*2, *i*7, *i*9, and *i*10 require cooling water (CW); tasks *i*1, *i*3, *i*5, and *i*8 require low-pressure steam (LPS); and tasks *i*4 and *i*6 require high-pressure steam (HPS). The maximum availabilities of cooling water and low- and high-pressure steam are 25, 40 and 20 kg/min, respectively.

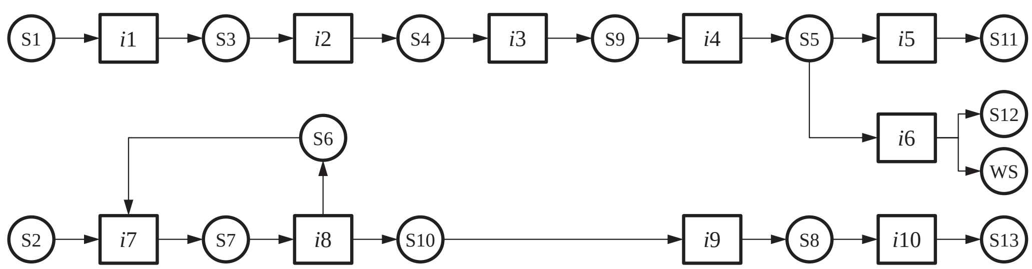

Figure 5.4 Recipe diagram for Example 2

We solve this example for the objective of revenue maximization. We use two scheduling horizons; Example 2a: H = 12 h and Example 2b: H= 14 h. Table 5.4 shows that for example 2b all models except the multi-grid model of Shaik and Floudas [82] perform equally well and can achieve an objective of \$16350 within comparable time. The SF model takes unusually long time to get this solution. This can be attributed to the large number of binary variables used by this model. In general for example 2b the multi-grid models took longer to get the optimal solution as compared to the single-grid models. This difference is more visible with example 2a, where there is an order of magnitude difference in the solution times of single- and multi-grid models. The poor performance of multi-grid formulations compared to the single grid models is, as also explained in Susarla et al. [206], because of the use of additional sequencing constraints if big-M type. However, for both scenarios, SLKs require fewer binary variables than all other models.

5.5.3 Example 3

This example is again taken from Maravelias and Grossmann [76]. This special characteristics of this example include sequence-dependent changeover times, shared storage tanks, and variable processing times. Figure 5.5 gives the recipe diagram. Tables 5.1 and 5.6 list the data. This example involves 2 units ($j1$ – $j2$), 6 tasks ($i1$ – $i6$), 8 materials ($s1$ – $s8$), and 2 shared storage tanks ($t1$ – $t2$). We solve this example for the objective of maximization of revenue for a scheduling horizon of 12 hrs.

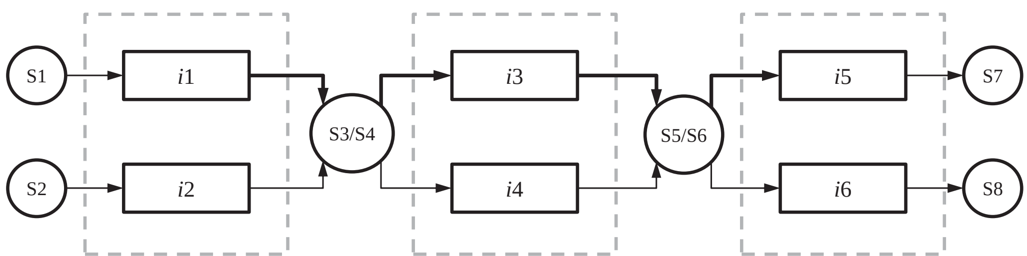

Figure 5.5 Recipe diagram for Example 3

For this example, we only implement MG apart from SLKs. Other models required small modifications to be able to be adopted for this example. Table 5.7 shows that SLKs clearly perform better than MG, as our models require fewer slots (6 vs. 9). Although the solution times for all are comparable, our models require fewer binary variables (40 vs. 108), continuous variables (353 vs. 514), constraints (561 vs. 1295), and nodes (390 vs. 1283).

Table 5.4 Batch size data for Example 2

	B_{ij}^U						α_{ij}						γ_{ij}						μ_{ri}					
	$j1$	$j2$	$j3$	$j4$	$j5$	$j6$	$j1$	$j2$	$j3$	$j4$	$j5$	$j6$	$j1$	$j2$	$j3$	$j4$	$j5$	$j6$	$j1$	$j2$	$j3$	$j4$	$j5$	$j6$
$i1$	5						2						3						2					
$i2$		8						1						4						2				
$i3$			6						1						4						3			
$i4$	5						2						3						2					
$i5$				8						2						8						4		
$i6$				8						2						4						3		
$i7$					3						4						5						4	
$i8$						4						2						5						3
$i9$					3						2						5						3	
$i10$						4						3						3						3

Table 5.5 Model and solution statistics for Example 2

Model	K	CPU time (s)	Nonzeros	Nodes	MILP	Binary variables	Continuous variables	Constraints
Example 2a: Revenue Maximization								
SLK1	**9**	**316**	**6964**	**41736**	**$13,000**	**128**	**979**	**2080**
SLK2	**9**	**335**	**6912**	**96757**	**$13,000**	**128**	**1009**	**2032**
SLK-sg	**9**	**11**	**3404**	**6105**	**$13,000**	**128**	**739**	**703**
CBMN	**9***	**3.4**	**2702**	**2404**	**$13,000**	**150**	**367**	**745**
SF	**9***	**2085**	**10147**	**240519**	**$13,000**	**240**	**588**	**2168**
MG	**9**	**15**	**8114**	**5315**	**$13,000**	**180**	**1009**	**2252**
Example 2b: Revenue Maximization								
SLK1	**8**	**4.5**	**5923**	**947**	**$16,350**	**112**	**865**	**1764**
SLK2	**8**	**5.7**	**6043**	**1302**	**$16,350**	**112**	**801**	**1778**
SLK-sg	**8**	**0.4**	**2861**	**320**	**$16,350**	**112**	**651**	**614**
CBMN	**8***	**0.14**	**2353**	**15**	**$16,350**	**130**	**323**	**655**
SF	**10***	**10000+**	**11353**	**435414**	**$16,350**	**270**	**655**	**2408**
MG	**8**	**0.3**	**6886**	**10**	**$16,350**	**160**	**903**	**2001**

Table 5.6 Batch size data for Example 3

	B_{ij}^{U}		B_{ij}^{L}		α_{ij}		β_{ij}		$\tau_{ii'}$					
	$j1$	$j2$	$j1$	$j2$	$j1$	$j2$	$j1$	$j2$	$i1.j1$	$i2.j1$	$i3.j2$	$i4.j2$	$i5.j1$	$i6.j1$
$i1$	5		2		0.5		0.4			0.2			0.1	0.5
$i2$	5		2		0.75		0.6		0.3				0.6	0.4
$i3$		3		1.2		1		1.333				0.3		
$i4$		3		1.2		1		1.333			0.2			
$i5$	5		2		0.5		0.4		0.2	0.6				0.5
$i6$	5		2		0.5		0.4		0.5	0.3			0.4	

Table 5.7 Model and solution statistics for Example 3

Model	K	CPU time (s)	Nonzeros	Nodes	MILP	Binary variables	Continuous variables	Constraints
Example 3: Revenue Maximization								
SLK1	**6**	**0.35**	**2101**	**171**	**\$5.020**	**40**	**353**	**571**
SLK2	**6**	**0.44**	**2189**	**390**	**\$5.020**	**40**	**333**	**561**
SLK-sg	**6**	**0.24**	**1513**	**97**	**\$5.020**	**80**	**265**	**352**
MG	**9**	**4.7**	**5571**	**1283**	**\$5.020**	**108**	**514**	**1295**

5.6 Summary

In this work, we utilize the concept of unit slots, originally proposed by Susarla et al., [206] to modify and extend the original formulation to consider utility handling constraints, sequence dependent changeover/setup times, and non-zero material transfer times. We further extend the model of Sundaramoorthy and Karimi [70] and present a new single-grid model based on process slots that can account for various resources, sequence-dependant changeover times, non-zero transfer times, and non-simultaneous material transfers. The model comparison shows that our models preserve their superiority among the equals. A critical point in all our models is that they do not require any additional binary variable to know the relative positions of the tasks to model the sequence-dependent changeover times, as commonly practiced in slot based formulations. Also, our models do not define any additional variables to model the non-zero material transfer times. One of the major contributions of this work is that it lays a clear understanding on the advantages and limitations of both single and multi-grid formulations. Our study highlights that dealing with the additional features such as non-zero transfer times and non-simultaneous material transfers, multi-grid approach offer higher flexibility to model time and handle related constraints. On the other hand, single-grid approach tends to become highly complex in allowing such features. Finally, a key outcome of this study is the importance of integrating resources in production scheduling and studying their effect on the process performance. This lays a clear direction for a detailed study of the effect of resources on the process performance not only on the process scale but on the plant scale. We explore this further in our next chapter.

6 INTEGRATED CAMPAIGN PLANNING AND RESOURCE ALLOCATION IN BATCH PLANTS[1, 2]

6.1 Introduction

Production plants operating in batch mode are highly common in chemical process industries, e.g. specialty chemicals, pharmaceuticals, food processing, and paints. Multiple products, multi-stage operations, fluctuating demands, and limited resources are a few of the typical characteristics for these industries. The degree of flexibility in process operations and the ease of adaptability to changing market scenarios forces such industries to operate in batch mode. Furthermore, the level of globalization requires these industries to streamline and re-design their supply chain operations. Also, because of the increasing market competition, companies are now facing an immense pressure to reduce the cost of finished goods. In this regard, optimization of manufacturing process through optimal resource allocation and lean operations offers a great potential to reduce cost of production in batch process industries. Thus, operational planning in such companies is highly important and so, is done frequently.

[1] Susarla, N., & Karimi, I. A. (2011). Integrated campaign planning and resource allocation in batch plants. Computers & Chemical Engineering, 35, 2990-3001.
[2] Susarla, N., & Karimi, I. A. (2010). Integrated campaign planning and resource allocation in batch plants. Presented in ESCAPE 2010, Ischia, Naples, Italy.

Operational planning in multi-product, multi-stage batch plants (MMBPs) is inherently complex, involves a plethora of decisions, and is usually for a horizon of 2 or 3 years. Two main considerations for planners in such plants are the frequent product changeovers and the long cleaning times involved. To minimize the cost and time needed for these changeovers, the plants are operated in campaign mode. A campaign usually consists of several batches of the same product or stage. However, long campaigns increase the inventory of products, which again incur cost. Thus, an optimal trade-off is required between campaign lengths and inventories. In addition to campaign lengths, other planning decisions include campaign sequencing, resource allocation, maintenance plan, and new product introductions (NPIs). Although planning in most of the companies is done by a dedicated planning department, it is a collaborative process (Figure 6.1). This is because planning process seeks inputs from several other departments of the company. The departments typically involved in the planning activity include process, maintenance, laboratory, sales, suppliers, and higher management. All these departments provide inputs like demand forecasts, maintenance plan, strategies for NPIs, projected market and business scenarios, and availability of resources such as human, equipment, utilities, and raw materials. Considering that most of these inputs are for future, the values for these inputs keep changing with time. To include these changes, the plan is reviewed regularly by all the departments. While reviewing and updating the plan, one of the main attentions is on the allocation of resources. This is mainly because the plant productivity depends greatly on the resource allocation profile. Productivity of the plant can either be expedited by allocating more resources or impeded by allocating insufficient resources. Utilizing this degree of freedom, planners analyze various operational scenarios and find the one that best suits the current business requirements. This is again based on several parameters such as plant utilization, operator over-time, and inventories.

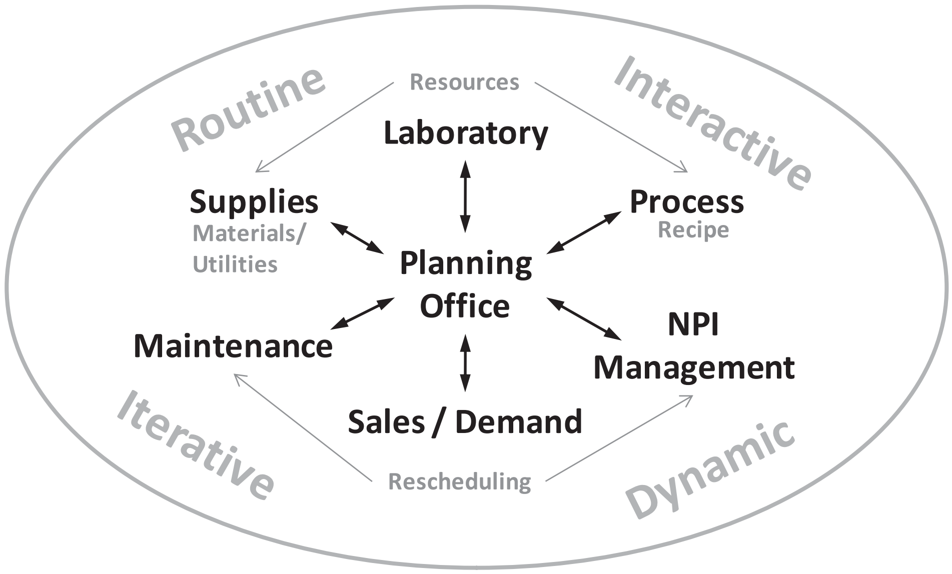

Figure 6.1 A schematic of collaborative planning in a pharmaceutical company

Now, we describe a typical scenario of industrial planning activity. Planners usually update a plan in the 1st week of each month with the actual production data. The unsatisfied demands (if any) are then carried forward and adjusted in the future. This updated plan is reviewed by in-plant departments such as process, maintenance, and laboratory in the 2nd week of the month. In this review, the in-plant departments verify and discuss the availability of various resources (human, equipment, and laboratory) for implementing the production plan. Also, various scenarios are evaluated based on different resource allocation profiles and maintenance plans. Based on this review, planners incorporate required changes and update the plan. In the 3rd week of the month, the latest plan is reviewed by the higher management of the company. Here, the management body evaluates the suitability of the proposed plan to the current business and market needs. Also, the potential new products' testing and the operational strategy are reviewed. Planners and the higher management then collaboratively evaluate more operational scenarios by tweaking campaign schedules, NPIs, and resource assignments/requirements. Finally, the changes warranted by the higher management are incorporated into the plan. In addition to the aforementioned monthly meetings, the plan is also reviewed bimonthly or quarterly by the suppliers and the sales department. In this meeting, raw material availabilities, supply, demand forecasts, and market scenarios are reviewed. This again demands changes in the plan. Evidently, the planning activity is highly frequent in MMBPs and requires sophisticated approaches. Our discussion with one such MMBP revealed that existing commercial tools are either less flexible in evaluating different scenarios or are too complex to be used by the planners. Thus, planners in the industry mostly use simple spreadsheets for planning. This is a time consuming process and may involve errors. On an average an experienced planner takes about 2 working days to complete one scenario of a feasible plan.

Our review of the existing literature in Chapter 2 highlights that only a few of the existing works consider resource scheduling constraints along with the production planning. Also, few works study the variation of productivity based on the resource allocation. Additionally, only a few models potentially allow the flexibility to generate a number of scenarios based on different resource allocation profiles and market conditions. This forms the basic motivation our study that we present in this chapter.

Our specific goal in this chapter is to develop a simple mathematical model (MILP) for the integrated problem of operational planning and resource allocation in MMBPs. We present a general framework to perform a scenario study based on the variation of productivity with different resource allocation profile. In addition, we allow several real life scenarios such as maintenance, NPIs, outsourcing of intermediates/products, safety stock limits, minimum campaign lengths, and sequence-dependent changeovers. With a trade-off between the mathematical complexity and the scope of problem, our framework comes handy for planners in the process industries.

6.2 Problem Statement

A multipurpose manufacturing facility (**F**) has *J* batch units (j = 1, 2, …, J). It handles *S* materials (s = 1, 2, …, S) including raw, intermediates, products, and wastes. For the current planning horizon, it has demand profiles with pre-specified due dates or delivery dates (DDs). Producing each product involves a known sequence of processing or production tasks (i). Let **IP** denote the set of all such tasks ($i \in$ **IP**). A recipe diagram, Figure 6.2 in this chapter [70, 206], gives detailed information on the various production tasks, materials, and resources.

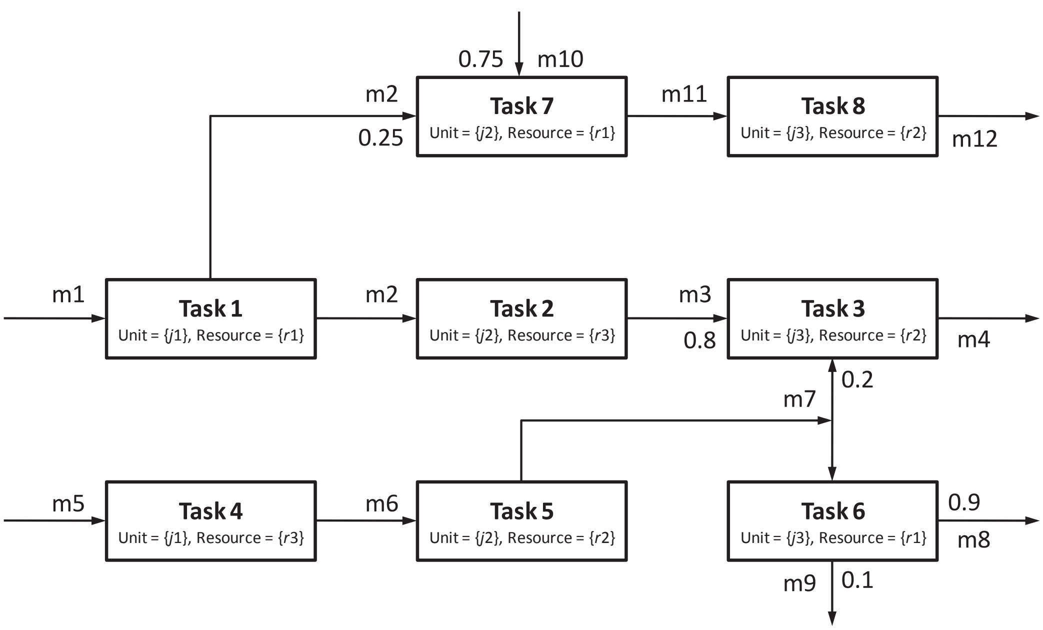

Figure 6.2 Recipe diagram for Example 1. r1 represents LP-steam, r2 represents HP-steam, r3 represents cooling water.

F uses the campaign mode of operation. In other words, it runs single-task campaigns on various units over time to make its products. Each campaign runs on a specific unit and produces a series of identical batches. Let b_{ij} denote the constant and known batch size of task i on unit j. Every batch requires some materials and resources. We use mass ratios (σ_{si} : s = 1, 2, …, S, $i \in$ **IP**; [206]) to quantify the amounts of materials involved in a single batch of a task i. $\sigma_{si} > 0$, if task i produces material s, and vice versa. One batch of task i on unit j produces $\sigma_{si}b_{ij}$ of material s.

Each unit j comprises an ordered series of equipment items such as mixers, reactors, crystallizers, driers, etc. that process each batch. Let pt_{ij} be the known total residence (processing) time of a batch of task i within unit j. During a campaign of identical batches, the time (cycle time or ct_{ij}) to produce one batch is shorter than pt_{ij} due to the staged configuration of equipment in each unit. ct_{ij} is usually pre-computed during the plant-fit of the process by the technical department based on the residence and cleaning times within each equipment item. Thus, the planners in the industry (and we in this work) treat ct_{ij} as a given parameter.

To ensure smooth operations, **F** may perform spot or routine maintenance on various units. Since a unit undergoing maintenance becomes unavailable for production, we define each such maintenance activity as a task on that unit and **IM** as the set of all planned maintenance tasks. **F** may also occasionally perform production trials of pre-specified durations to produce new products. As with maintenance, we model each such activity as a distinct task, and define **IN** as the set of all stipulated NPI (New Product Introduction) tests. Then, we define $\mathbf{I} = \mathbf{IP} \cup \mathbf{IM} \cup \mathbf{IN}$ and $\mathbf{I}_j = \{i \mid$ unit j can perform task $i \in \mathbf{I}\}$. We model the known durations for maintenance and production trials for NPIs as processing time pt_{ij}. Now, the time of performing maintenance and trials of NPIs are pre-specified and so, are fixed. Thus, planners may generate different production scenarios based on different plans for maintenance and NPIs.

Let us say that the various tasks (production, idling, maintenance, NPI) in **F** require R resources ($r = 1, 2, \ldots, R$), hence we define $\mathbf{I}_r = \{i \in \mathbf{I} \mid \text{task } i \text{ needs resource } r\}$.

With this, the planning problem addressed in this paper can be stated as follows.

Given:

1. Production units, raw materials, intermediates, products, tasks, and recipes
2. Planning horizon (H), batch sizes, processing times, cycle times, and sequence-dependent changeover or cleaning times
3. Demand forecasts, safety stock limits, and initial inventories
4. Planned or anticipated maintenance schedule
5. Product prices and raw material costs
6. Resources and their availability profiles

Determine:

1. Optimal production plan (campaigns, schedules, and batches) and corresponding KPI
2. Material stock and resource allocation profiles

Assuming:

Deterministic scenario

Aiming for:

Maximum Revenue

6.3 MILP Formulation

Unless otherwise indicated, an index takes all its legitimate values in all the expressions or constraints in our formulation. We use [60, 144] the known order delivery dates (DD_0 (= 0) < DD_1 < DD_2 < DD_3 < ...) to segment the planning horizon

[0, H] into T non-uniform intervals (t = 1, 2, …, T) of length $h_t = DD_t - DD_{(t-1)}$. Furthermore, we split every interval t on each unit j into K_j (k = 1, 2, …, K_j) ordered and contiguous slots (Figure 6.3) of unknown and variable lengths [206]. This gives the flexibility to allocate a different number (K_j) of slots to each unit j. K_j will vary based on the number of tasks that unit j may be able to perform, i.e. $|\mathbf{I} : i \in \mathbf{I}_j|$. This is useful in cases where the units vary significantly in terms of their "multi-purpose" processing abilities (e.g. dedicated units).

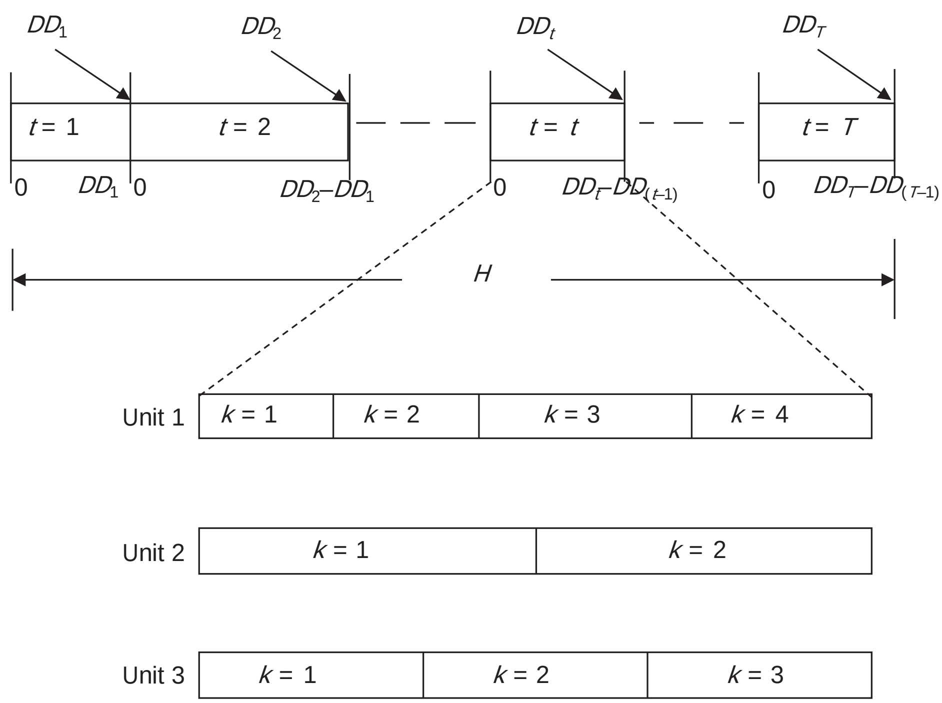

Figure 6.3 Design of unit-slots [13, 206]

Let T_{jkt} [k = 0, 1, 2, …, K_j; $T_{j0t} \geq 0$; $T_{jK_jt} \leq h_t$] denote the end time of slot k on unit j in interval t. Slot 0 refers to the slot just before interval t or the last slot of interval (t–1). The slots are ordered and contiguous in time, and the length of slot k is $(T_{jkt} - T_{j(k-1)t})$.

6.3.1 Campaign allocation

For the ease in writing, we use $\mathbf{C}_{ijt}$ to denote a campaign of task i on unit j during interval t. We allocate exactly one single-task campaign to each slot of all units by using the following binary variable and constraint.

$$x_{ijkt} = \begin{cases} 1 & \text{if } \mathbf{C}_{ijt} \text{ runs in slot } k \\ 0 & \text{otherwise} \end{cases} \qquad i \in \mathbf{I}_j,\ 1 \le j \le J,\ 1 \le k \le K_j,\ 1 \le t \le T$$

$$\sum_{i \in \mathbf{I}_j} x_{ijkt} = 1 \qquad 1 \le j \le J,\ 1 \le k \le K_j,\ 1 \le t \le T \tag{6.1}$$

For the ease in writing our constraints, we use $x_{ij0(t+1)}$ also as an alias for x_{ijK_jt}. As we know the last task processed by unit j before the start of our planning horizon, we fix x_{ij01} appropriately to reflect that.

Campaign changeovers or cleaning times ($\tau_{ii'}$) on a unit are usually dependent on the sequence of tasks. To include appropriate and sufficient changeover times, we must identify the sequence of campaigns on each unit. To this end, we use the following 0-1 continuous variable [13, 62].

$$y_{ii'jkt} = \begin{cases} 1 & \text{if } \mathbf{C}_{ijt} \text{ is in slot } k \text{ and } \mathbf{C}_{i'jt} \text{in slot } (k+1) \text{ of interval } t \\ 0 & \text{otherwise} \end{cases}$$

$$i, i' \in \mathbf{I}_j,\ 1 \le j \le J,\ 0 \le k < K_j,\ 1 \le t \le T$$

Clearly, $y_{ii'jkt} = x_{ijkt} x_{i'j(k+1)t}$, and we can linearize as follows.

$$\sum_{i' \in I_j} y_{ii'jkt} = x_{ijkt} \qquad i \in \mathbf{I}_j,\ 1 \le j \le J,\ 0 \le k < K_j,\ 1 \le t \le T \tag{6.2a}$$

$$\sum_{i \in I_j} y_{ii'jkt} = x_{i'j(k+1)t} \qquad i \in \mathbf{I}_j,\ 1 \le j \le J,\ 0 \le k < K_j,\ 1 \le t \le T \tag{6.2b}$$

$$y_{iij0(T+1)} = x_{ij0(T+1)} \qquad i \in \mathbf{I}_j,\ 1 \le j \le J \tag{6.2c}$$

Based on what unit j is doing at time zero, we fix x_{ij01} appropriately. Note that $y_{iij0t} = 1$, if $\mathbf{C}_{ijt}$ continues from interval t to $(t+1)$. Equation 6.2c forces the last campaign in the horizon to always continue.

To avoid cleaning and changeovers between campaigns and resulting costs, times, and wastes, we would like to restrict the number of campaigns of a task to one in each interval. However, we could have more slots than campaigns in an interval. Therefore, we allow only the last campaign in each interval to have multiple slots.

$$y_{iij(k+1)t} \geq y_{iijkt} \qquad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq k < K_j,\ 1 \leq t \leq T \qquad (6.3)$$

Equation 6.3 ensures that if any campaign spans more than one slot in any interval, then it spans all the remaining slots in that interval, thus becoming the last campaign.

Finally, we detect the existence of $\mathbf{C}_{ijt}$ by using the following 0-1 continuous variable.

$$z_{ijt} = \begin{cases} 1 & \text{if } \mathbf{C}_{ijt} \text{ exists} \\ 0 & \text{otherwise} \end{cases} = x_{ij1t} + \sum_{k=2}^{K_j} [x_{ijkt} - y_{iij(k-1)t}] \qquad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq t \leq T \qquad (6.4)$$

Now, the decisions pertaining to maintenance and trials for NPIs are usually taken in consultation with the respective departments (maintenance, process, R&D, etc.). Thus, the planners often have less flexibility in changing the schedules for such operations. In this work, we assume that the schedules for the maintenance and trials for NPIs are supplied a priori to the planners. To consider such schedules along with routine production activities in a given interval, we fix appropriate values for z_{ijt}. This is unlike Sundaramoorthy & Karimi [144], where they modelled the production trials for NPIs as just extra production tasks.

6.3.2 Campaign / slot lengths

The plant operations may require that each campaign must always produce some minimum batches before it ends. Let n_{ij}^{L} denote this minimum number of batches that $\mathbf{C}_{ijt}$ must produce. For maintenance and NPIs, we specify n_{ij}^{L} = 1. Now, to ensure this requirement, we define two integer variables n_{ijt} ($i \in \mathbf{I}_j$, $1 \le j \le J$, $0 \le t \le T$) and Δn_{ijt} ($i \in \mathbf{I}_j$, $1 \le j \le J$, $1 \le t \le T$) as follows. Δn_{ijt} is the number of batches that $\mathbf{C}_{ijt}$ produces during interval t.

Clearly, if $\mathbf{C}_{ijt}$ does not exist, then it cannot produce any batch during t.

$$\Delta n_{ijt} \le n_{ijt}^{U} z_{ijt} \qquad i \in \mathbf{I}_j,\ 1 \le j \le J,\ 1 \le t \le T \tag{6.5}$$

where, n_{ijt}^{U} is the most batches that $\mathbf{C}_{ijt}$ can produce in interval t. For maintenance and NPIs, we specify n_{ijt}^{U} = 1. Since the interval lengths may vary, this limit will change proportionally. Now, for n_{ijt}, we have three scenarios.

1. $\mathbf{C}_{ijt}$ does not exist. Then, we define $n_{ijt} = 0$.
2. $\mathbf{C}_{ijt}$ continues into interval (t+1). Then, we define n_{ijt} as the total number of batches that it completes from its start to the end of interval t. Note that it may have begun in interval t or before.
3. $\mathbf{C}_{ijt}$ ends in interval t. Then, we define $n_{ijt} = 0$.

The above definition of n_{ijt} gives us,

$$n_{ijt} \le n_{ij(t-1)} + \Delta n_{ijt} \qquad i \in \mathbf{I}_j,\ 1 \le j \le J,\ 1 \le t < T \tag{6.6a}$$

$$n_{ijt} \ge n_{ij(t-1)} + \Delta n_{ijt} - n_{ij}^{U}(1 - y_{iij0(t+1)}) \qquad i \in \mathbf{I}_j,\ 1 \le j \le J,\ 1 \le t < T \tag{6.6b}$$

$$n_{ijt} \le n_{ij}^{U} y_{iij0(t+1)} \qquad i \in \mathbf{I}_j,\ 1 \le j \le J,\ 1 \le t \le T \tag{6.6c}$$

where, n_{ij}^{U} is the most batches that $\mathbf{C}_{ijt}$ could produce in the entire planning horizon. Note that $n_{ij0} = 0$, if a campaign is not running at time zero. Otherwise, it equals the number of batches that the current campaign has produced by time zero. $n_{ij(t-1)} + \Delta n_{ijt}$ gives the batches that $\mathbf{C}_{ijt}$ has completed by the end of interval t. If $\mathbf{C}_{ijt}$ ends in interval t, then it must have produced the minimum required batches.

$$n_{ij(t-1)} + \Delta n_{ijt} \geq n_{ij}^{L}[z_{ijt} - y_{iij0(t+1)}] \qquad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq t \leq T \qquad (6.7)$$

Note that eq. 6.7 is relaxed, whenever $\mathbf{C}_{ijt}$ continues into interval (t+1). This allows $\mathbf{C}_{ijt}$ to produce the minimum required batches by interval (t+1) or beyond.

Having modelled the batches produced by $\mathbf{C}_{ijt}$, we now ensure that it has sufficient time to produce them. To this end, we define a continuous variable RL_{ijkt} (run length; $i \in \mathbf{I}_j$, $1 \leq j \leq J$, $1 \leq k \leq K_j$, $1 \leq t \leq T$) as the time allocated to $\mathbf{C}_{ijt}$ during slot k. If task i is not allocated to slot k, then its run length must be zero.

$$RL_{ijkt} \leq h_t x_{ijkt} \qquad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq k \leq K_j,\ 1 \leq t \leq T \qquad (6.8)$$

Since the sum of run lengths must exceed the slot length, we write,

$$T_{jkt} - T_{j(k-1)t} \geq \sum_{i \in \mathbf{I}_j} RL_{ijkt} \qquad 1 \leq j \leq J,\ 1 \leq k \leq K_j,\ 1 \leq t \leq T \qquad (6.9)$$

Now, to compute the total time allocated to a campaign, we define a current campaign length CL_{ijt} for $i \in \mathbf{I}_j$, $1 \leq j \leq J$, $0 \leq t \leq T$ analogous to n_{ijt} defined earlier. Similar to n_{ijt}, we have the following three scenarios for CL_{ijt}.

1. If $\mathbf{C}_{ijt}$ does not exist, then we define $CL_{ijt} = 0$.
2. If $\mathbf{C}_{ijt}$ continues into interval (t+1), then we define CL_{ijt} as the total time that $\mathbf{C}_{ijt}$ has used since it first began to the end of interval t.
3. If $\mathbf{C}_{ijt}$ ends in interval t, then we define $CL_{ijt} = 0$.

Then, analogous to eq. 6.6, we have,

$$CL_{ijt} \leq DD_t \cdot y_{iij0(t+1)} \qquad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq t \leq T \qquad (6.10a)$$

$$CL_{ijt} \leq CL_{ij(t-1)} + \sum_{k=1}^{K_j} RL_{ijkt} \qquad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq t \leq T \qquad (6.10b)$$

$$CL_{ijt} \geq CL_{ij(t-1)} + \sum_{k=1}^{K_j} RL_{ijkt} - DD_t(1 - y_{iij0(t+1)}) \quad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq t \leq T \qquad (6.10c)$$

If a campaign is not running at time zero, then $CL_{ij0} = 0$. Otherwise, CL_{ij0} equals its current campaign length at time zero.

Knowing the batches produced by and total time allocated to a campaign at various intervals, we now ensure that campaign length at the end of each interval is sufficient to produce all the batches until that interval with an appropriate consideration of the sequence–dependent changeover times. We define campaign "start" as the time at which the campaign begins processing its first batch, and the end as the time at which the following campaign begins processing its first batch. Thus, in addition to the time required to produce all the batches, changeover time for the following campaign and any idle time are also included in a campaign's required length. If $\mathbf{C}_{ijt}$ ends in interval t, then C_{ijt} must exceed the total time required to produce $n_{ij(t-1)} + \Delta n_{ijt}$ batches plus an appropriate changeover time for the subsequent campaign.

$$CL_{ij(t-1)} + \sum_{k=1}^{K_j} RL_{ijkt} \geq (pt_{ij} - ct_{ij})\left[z_{ijt} - y_{iij0(t+1)}\right] + (n_{ij(t-1)} + \Delta n_{ijt})ct_{ij}$$

$$+ \sum_{k=1}^{K_j - 1} \sum_{i' \in \mathbf{I}_j} \tau_{ii'} y_{ii'jkt} + \sum_{i' \in \mathbf{I}_j} \tau_{ii'} y_{ii'j0(t+1)} \qquad i \in \mathbf{I}_j,\ 1 \leq j \leq J,\ 1 \leq t \leq T \quad (6.11)$$

6.3.3 Inventories

If $\mathbf{C}_{ijt}$ exists, then each of its batches must have sufficient precursor materials before it can begin. In other words, we must check the inventory of precursor materials at the start of every batch. We argue that this level of rigor makes the planning problem too complex, and is unwarranted. Thus, we check inventories only at interval (rather than slot) ends.

Let D_{st} be the demand of material state s, S_{st} ($\le D_{st}$) denote the amount shipped from the plant, and Q_{st} ($\le Q_{st}^U$) denote the net inventory left at the end of interval t. Furthermore, let OS_{st} denote the amount outsourced during interval t. Then, from Sundaramoorthy & Karimi [144], we write the inventory balance as,

$$Q_{st} = Q_{s(t-1)} + OS_{st} + \sum_{j} \sum_{i \in \mathbf{I}_j \cap \mathbf{IP} \cap \mathbf{IN}} \sigma_{si} \Delta n_{ijt} b_{ij} - S_{st}$$

$$1 \le s \le S,\ 1 \le t \le T,\ \sigma_{si} \ne 0 \qquad (6.12)$$

Q_{s0} is the known initial inventory of state s at time zero. Note that eq. 6.12 allows us to align the tasks in different intervals based on the recipe by checking the availability of the precursor materials at the beginning of each interval.

It is sometimes not possible for a plant to meet the demand in an interval. We assume that this demand is carried over to the next interval. Then, this demand overflow (DO_{st}) at the end of interval t is given by,

$$DO_{st} \ge DO_{s(t-1)} + D_{st} - S_{st} \qquad 1 \le s \le S,\ 1 \le t \le T \qquad (6.13)$$

SS_{s0} is the pre-specified demand carry-over at time zero. To prevent shipments exceeding the orders, we use,

$$S_{st} \le D_{st} + DO_{st} \qquad 1 \le s \le S,\ 1 \le t \le T \qquad (6.14)$$

The plant may maintain a safety stock (SS_{st}) for a material state s to ensure a continuous supply of raw materials and products and to guard against uncertain demand. We penalize, if an inventory falls below this safety stock at the end of an interval. We compute this violation (SSV_{st}) by,

$$SSV_{st} \geq SS_{st} - Q_{st} \qquad 1 \leq s \leq S,\ 1 \leq t \leq T \qquad (6.15)$$

6.3.4 Campaign timings within intervals

As discussed earlier, each batch will need certain amounts of precursor materials, before it begins. As long as any task that produces such a precursor material occurs in a previous interval, the inventory checks imposed in the previous section will ensure this. However, the same cannot be guaranteed within an interval. Thus, we adopt an approximate treatment for this as proposed by Sundaramoorthy and Karimi [144]. Let i' denote a task that produces one or more precursor materials for a task i. If both i and i' occur in the same interval, then we demand that task i must begin after i'. If $\delta_{ii'}$ denotes a pre-specified time delay between the starts of i and i', then we require,

$$T_{jkt} \geq T_{j'k't} + \delta_{ii'} - h_t\Big(2 - \sum_{\substack{i'' \in \mathbf{I}_j \cap \mathbf{I}_r \\ i'' \neq i}} y_{i''ijkt} - \sum_{\substack{i'' \in \mathbf{I}_j \cap \mathbf{I}_r \\ i'' \neq i'}} y_{i''i'j'k't}\Big) \qquad i \in \mathbf{I}_j \cap \mathbf{IP},\ i' \in \mathbf{I}_{j'} \cap \mathbf{IP},$$

$$0 \leq k < K_j,\ 0 \leq k' < K_j,\ k \neq k',\ 1 \leq j \neq j' \leq J,\ \exists s: \sigma_{si} < 0,\ \sigma_{si'} > 0,\ 1 \leq t \leq T \qquad (6.16)$$

6.3.5 Resources

All chemical plants in general and batch plants in particular require several resources other than raw materials and processing units for production. These include utilities such as electricity, steam, chilled water, etc. and others such as human, indirect materials (e.g. catalysts), storage, laboratory, tools, parts, auxiliary equipment, etc. The availability and allocation of these resources directly impact the plant productivity. For instance, limited availability of operators may affect the processing and cycle times of batches in a labour-intensive batch plant. Thus, their allocation among various competing tasks becomes crucial. Without a proper consideration of resource allocation and availability, a given plan either underestimate or overestimate the total production, or may even be infeasible. Because the planner may not have full or current information on all resources and their limits, the initial production plan is typically reviewed by various other departments (maintenance, process, laboratory, higher management, etc.) for approval. In this work, we modify the approach of Susarla et al. [206] to monitor the usage of various resources within each time interval and ensure that they do not exceed the availabilities at any time.

Let the plant production involve R resources (r = 1, 2, …, R). We assume that resource usage is constant over time for each task during a campaign. Let γ_{ijr} denote the usage of resource r by task i on unit j. Now, similar to K_j, we split each interval t on each such resource r into K_r (k = 1, 2, …, K_r) ordered and contiguous slots of unknown and variable lengths.

$$T_{rkt} \geq T_{r(k-1)t} \qquad 2 \leq k < K_r \tag{6.17}$$

where T_{rkt} (r = 1, 2, …, R; k = 0, 2, …, K_r; T_{r0t} = 0; $T_{rK_rt} = h_t$) is the time at which slot (k+1) begins. Now, consider a slot k' on resource r during interval t and define the following binary variable.

$$u_{rijkt} = \begin{cases} 1 & \text{if task } i \text{ unit } j \text{ uses resource } r \text{ during slot } k \text{ of resource } r \text{ in interval } t \\ 0 & \text{otherwise} \end{cases}$$

$$1 \leq r \leq R,\ 1 \leq j \leq J,\ 1 \leq k \leq K_r,\ 1 \leq t \leq T$$

Because of the multi-grid time approach, several slots on resource r may be required for each campaign i that consumes the resource on unit j in interval t. Also, when a resource is not consumed all the related binary variables must be set to zero. Thus, we write the following,

$$\sum_{k=1}^{K_r} u_{rijkt} \geq z_{ijt} \qquad i \in \mathbf{I}_j \cap \mathbf{I}_r,\ 1 \leq r \leq R,\ 1 \leq j \leq J,\ 1 \leq t \leq T \tag{6.18a}$$

$$\sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \sum_{k=1}^{K_r} u_{rijkt} \leq K_r \sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} z_{ijt} \qquad 1 \leq r \leq R,\ 1 \leq j \leq J,\ 1 \leq t \leq T \tag{6.18b}$$

Again, multiple units may consume a resource parallel. So, this usage of resource r must be appropriately aligned with the start and end time of C_{ijt}.

$$T_{r(k'-1)t} \geq T_{j(k-1)t} - h_t (2 - u_{rijk't} - x_{ijkt})$$

$$i \in \mathbf{I}_j \cap \mathbf{I}_r,\ 1 \leq k \leq K_j,\ 1 \leq k' \leq K_r,\ k \leq k',\ 1 \leq r \leq R,\ 1 \leq j \leq J,\ 1 \leq t \leq T \tag{6.19a}$$

$$T_{rk't} \leq T_{jkt} + h_t (2 - u_{rijk't} - x_{ijkt})$$

$$i \in \mathbf{I}_j \cap \mathbf{I}_r,\ 1 \leq k \leq K_j,\ 1 \leq k' \leq K_r,\ k \leq k',\ 1 \leq r \leq R,\ 1 \leq j \leq J,\ 1 \leq t \leq T \tag{6.19b}$$

Note that Equations 6.17–6.19 only allow us to estimate the maxima of the resource usage profile. The consumption of any resource at any given instance must not exceed its maximum availability ($U_{rt}^{\max}$). This gives us the following,

$$\sum_{i \in \mathbf{I}_j \cap \mathbf{I}_r} \sum_{j} \gamma_{ijr} u_{rijkt} \leq U_{rt}^{\max} \qquad 1 \leq k \leq K_r,\ 1 \leq t \leq T,\ 1 \leq r \leq R \tag{6.20}$$

6.3.6 Planning objective and variable bounds

Two objectives have been used widely in the planning literature – revenue and profit maximization. The most preferred objective in planning process is the maximization of revenue through sales.

$$\max \text{revenue} = \sum_{s} \sum_{t} p_s S_{st} \tag{6.21}$$

where, p_s is the price of material s.

We consider that production process includes costs for material procurement (cm_s), changeover ($cc_{ii'}$), processing (cp_{ijt}), maintenance (cmt_{jt}) and NPI (cnp_{jt}), and material (intermediate and product) storage (ch_{st}). In addition, to minimize the violation of the safety stock levels and demand violation, we add penalties (csv_{st} and cdv_{st}, respectively) for each. This gives us the following objective for the profit.

$$\max \text{Profit} = \sum_s \sum_t p_s S_{st} - \text{cost} \tag{6.22}$$

where, $\text{cost} = \sum_s cm_s (Q_{s0} + \sum_t OS_{st}) + \sum_t \sum_k \sum_j \sum_{i' \in \mathbf{I}_j} \sum_{i \in \mathbf{I}_j} cc_{ii'} y_{ii'jkt} + \sum_t \sum_j \sum_{i \in \mathbf{I}_j} cp_{ijt} \Delta n_{ijt}$

$$+ \sum_t \sum_j \sum_{i \in \mathbf{I}_j \cap \mathbf{IM}} cmt_{jt} pt_{ij} z_{ijt} + \sum_t \sum_j \sum_{i \in \mathbf{I}_j \cap \mathbf{IN}} cnp_{jt} pt_{ij} z_{ijt} + \sum_s \sum_t ch_{st} Q_{st} + \sum_s \sum_t cdv_{st} DO_{st}$$

$$+ \sum_s \sum_t csv_{st} SSV_{st}$$

Appropriate bounds on the variables are known to improve the solution time. So, we specify the following bounds for the continuous variables. All variables are defined as non-negative variables, and the upper bounds are $y_{ii'jkt} \le 1$, $z_{ijt} \le 1$, $\Delta n_{ijt} \le n_{ijt}^U$, $RL_{ijkt} \le h_t$, $CL_{ijt} \le DD_t$, $T_{jkt} \le h_t$, $T_{rkt} \le h_t$, $Q_{st} \le Q_{st}^U$, $OS_{st} \le OS_{st}^U$.

This completes our model (pSK, Equations 6.1-6.20) for operational planning.

6.4 Numerical Evaluation

To study the performance of our model, we consider two examples and various operational scenarios for planning. For model implementation, we used CPLEX 12 (with default options)/GAMS 23.2 on a Dell precision T5500 workstation with Intel® Xeon® 2 x 2 GHz CPUs, 4 GB RAM, running Windows 7 Professional 64-bit operating system. For all examples, we consider same number of slots ($\mathbf{K}_j$) in all units.

6.4.1 Example 1

This example was presented by Sundaramoorthy and Karimi [144]. We modify the problem data to suit the current implementation. The example considers a facility *F* involving 8 tasks (*i*1-*i*8), 3 units (*j*1-*j*3), 12 materials (*s*1-*s*12), and 3 resources (LP-steam, HP-steam, and cooling water). *j*1 can perform *i*1 and *i*4, *j*2 can perform *i*2, *i*5, and *i*7, and *j*3 can perform *i*3, *i*6, and *i*8. *i*1, *i*6 and *i*7 require LP-steam, *i*3, *i*5, and *i*8 require HP-steam, and *i*2 and *i*4 require cooling water as additional resources apart from equipment and raw materials. *F* produces 3 products *s*4, *s*8, and *s*12 along with a by-product (waste) *s*9 consuming 3 raw materials *s*1, *s*5, and *s*10. Figure 6.2 gives the recipe diagram and Table 6.1 lists the new data.

We consider a case for F where the maximum availability of LP-steam is 60 mu/h, HP-steam is 40 mu/h, and cooling water is 60 mu/h. In our implementation, we assume a constant value $\delta_{ii'}$ = 30. The safety stock limits for the products are fixed as 2500 mu for *s*4, [1500, 2000] mu for *s*8, and 1000 mu for *s*12. There is no initial inventory available for any material other than the basic raw materials. Also, we assume that all units are free at the start of the planning horizon.

We solve this example for a planning horizon of 1 year with 12 time intervals of 30 days (720 h) each. We use Gantt charts to represent the plan given by our model. The horizontal rectangular bar represents the campaign of a task (indicated as a label). The campaigns shown include the changeover times and $\delta_{ii'}$. Now, we develop three different operational scenarios for planning in *F*. We solve all the scenarios for 10000 CPUs. The model statistics for all three cases are given in Table 6.2.

Table 6.1 Data for Example 1

tasks i	units j	min number of batches n_{ij}^L	max number of batches n_{ij}^U	batch size (mu)	processing time (h)	cycle time (h)	resource r	resource consumption (mu)
1	1	30	100	150	6	4	1	15
2	2	20	100	150	5	5	3	20
3	3	25	80	80	4	3	2	10
4	1	25	100	100	6	4	3	10
5	2	20	100	150	5	5	2	20
6	3	20	100	100	4	3	1	20
7	2	25	80	80	7	6	1	15
8	3	30	100	100	6	6	2	15

Table 6.2 Product demands and costs for Example 1

material	demand (mu) in interval *t*											
m	1	2	3	4	5	6	7	8	9	10	11	12
4	4000	5500	6000	4500	3000	2500	3500	4000	3500	2500	4500	5000
8	3000	2500	6500	3500	4000	2500	2000	3000	5000	4500	2500	3500
12	2500	3000	1500	2000	4500	3000	4000	2500	1500	3500	4000	3000

material *m*	revenue ($/mu)	safety stock target (mu)	cost of demand violation ($/mu)	cost of safety stock violation ($/mu)
4	1.4	2500	1	0.2
8	1	1500	1	0.2
12	1.2	1000	1	0.2

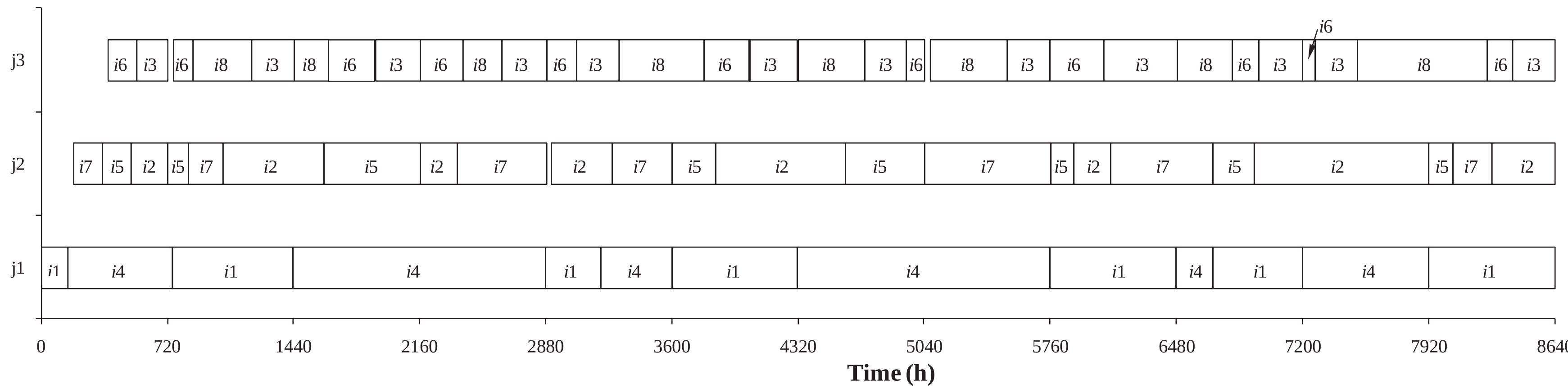

Figure 6.4 Plan for Scenario-1of Example 1

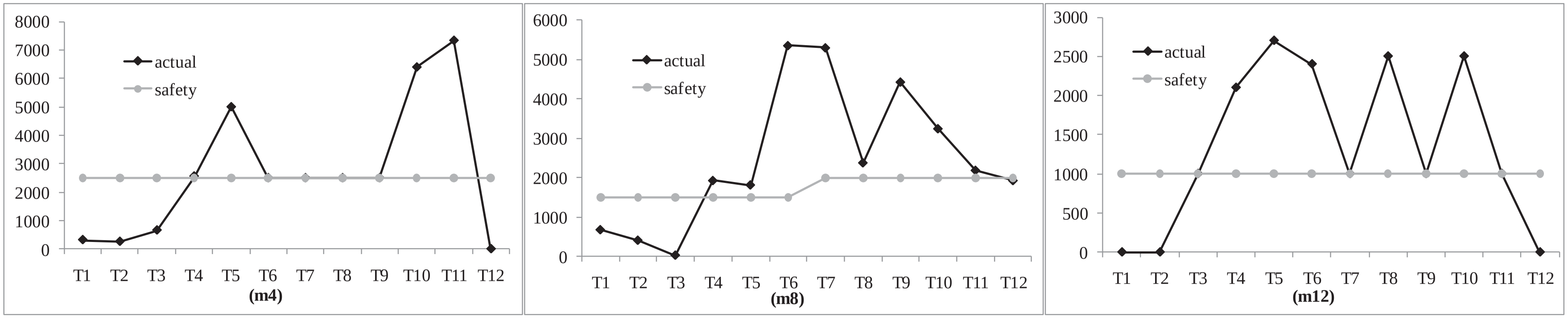

Figure 6.5 Product inventories for Scenario-1of Example 1

In scenario 1, we only consider production operations along with the resources. As this example did not involve any maintenance and NPIs, we fixed all the associated binary variables to zero. We used the objective of maximizing revenue along with penalties for the violation of safety stock and not meeting the demand. The problem consists of 1248 binary variables and 9225 constraints. An objective of $158113 was obtained. Figure 6.4 shows the Gantt chart for this solution and Figure 6.5 show the product inventories.

Now, the upper management may want to include trials for a few potential new products. Also, the maintenance department may want to perform overhauling of some of the equipment. For this, the planner needs to change the complete plan to incorporate desired provisions for maintenance and NPIs. So, in scenario 2, 3 maintenance and 4 new product trials are considered. This scenario includes maintenance for $j1$, $j2$, and $j3$ in interval $t3$, $t6$, and $t9$, respectively. Also, NPI trials during $t4$, $t5$, $t8$, and $t10$ on $j3$, $j1$, $j2$, and $j1$, respectively, are included. For this, we fix appropriate values for z_{ijt}. Also, we modified the objective to include costs for maintenance and NPI trials. An objective of $153841 was obtained. Figures 6.6 shows the Gantt chart and Figure 6.7 show product inventories. Note that the number of binary variables for scenario 2 is more than in scenario 1. This is because we fixed the binary variable related to maintenance and NPIs to zero in scenario 1.

Again, the solution obtained in scenario 2 may not satisfy all the stake holders of the plan. Also, scenario 2 does not allow the possibility of outsourcing intermediates or products. So, before making a decision the management may wish to see a few more scenarios. For this, we develop scenario 3 with different plan for maintenance and NPI trials. Here, we allow the possibility of outsourcing one of the intermediates ($s7$).

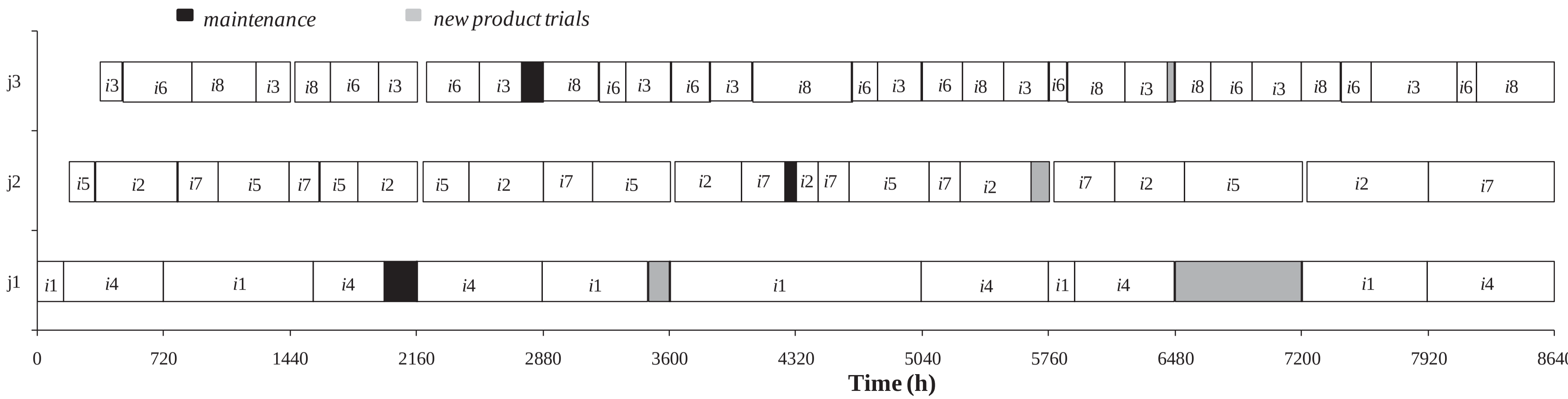

Figure 6.6 Plan for Scenario-2 of Example 1

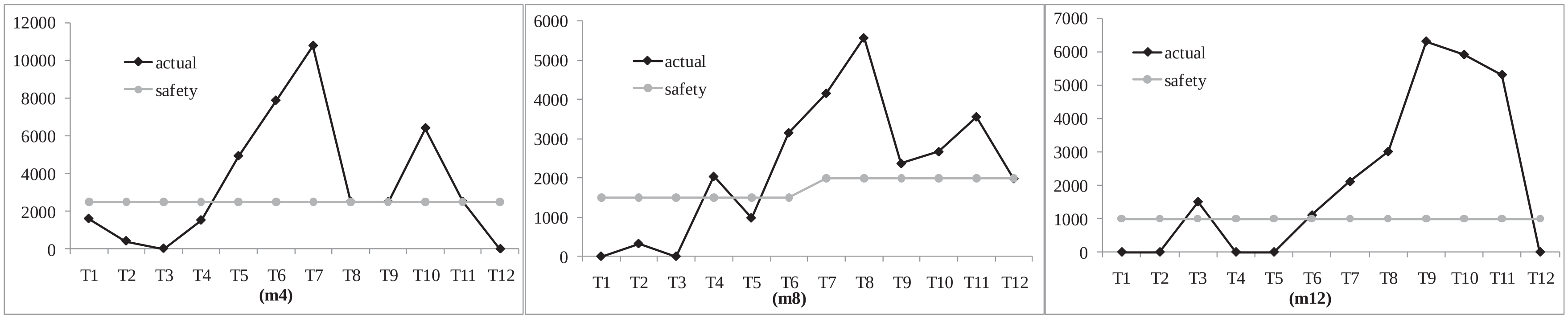

Figure 6.7 Product inventories for Scenario-2 of Example 1

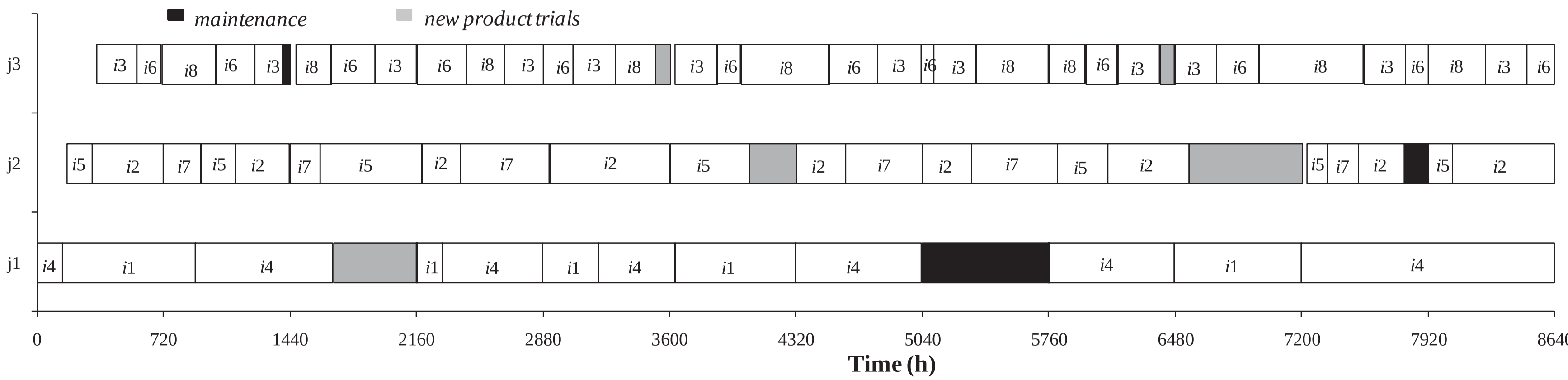

Figure 6.8 Plan for Scenario-3 of Example 1

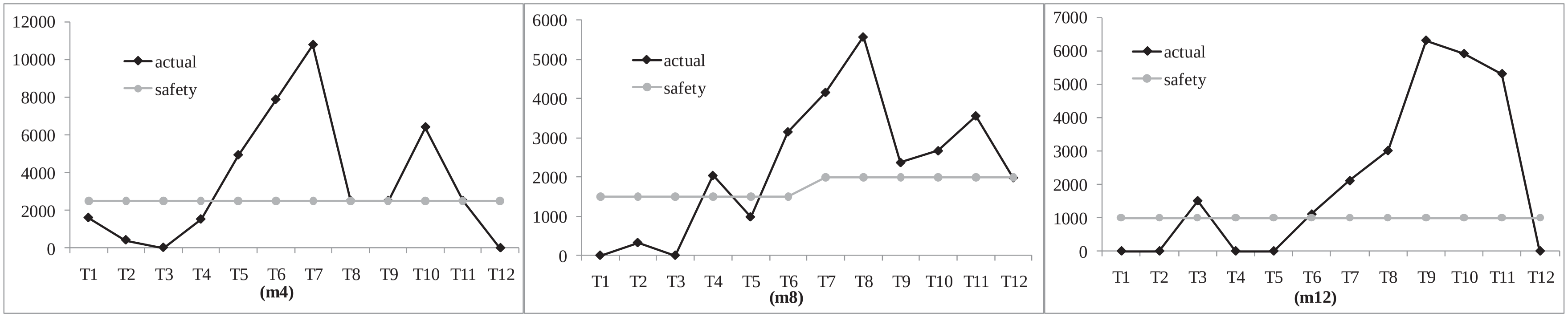

Figure 6.9 Product inventories for Scenario-3 of Example 1

Table 6.3 Data for Example 2

tasks i	units j	min number of batches n_{ij}^{L}	max number of batches n_{ij}^{U}	batch size (mu)	processing time (h)	cycle time (h)	resource r	resource consumption (mu)
1	1	35	150	150	6	4	1	10
2	2	30	100	100	5	5	3	15
3	3	35	120	80	4	3	2	10
4	4	20	150	100	5	4	1	15
5	1	40	170	150	6	4	2	20
6	2	25	150	200	5	5	4	15
7	3	45	80	80	4	3	1	20
8	2	20	200	150	7	6	2	25
9	3	25	180	100	6	6	3	20
10	4	30	100	100	6	5	4	25
maintenance task	1	1	1	-	40	-	-	-
	2	1	1	-	55	-	-	-
	3	1	1	-	35	-	-	-
	4	1	1	-	50	-	-	-
npi trials	1	1	1	-	80	-	-	-
	2	1	1	-	75	-	-	-
	3	1	1	-	65	-	-	-
	4	1	1	-	70	-	-	-

Table 6.4 Product demands and costs for Example 2

material	demand (mu) in interval *t*											
m	1	2	3	4	5	6	7	8	9	10	11	12
5	2500	1500	4500	5000	2000	3500	5000	4000	1500	3500	5500	6000
10	1000	1500	4500	2500	1000	1500	4500	2500	1000	1500	4500	2500
14	4000	2000	3500	1500	2500	4500	3000	2000	2500	3000	4000	3500

material	demand (mu) in interval *t*											
m	13	14	15	16	17	18	19	20	21	22	23	24
5	3500	1000	2500	4000	2000	1500	3500	4500	2500	2000	5000	6000
10	2000	2500	4000	3500	1500	2500	3500	2000	2500	3000	2500	4000
14	2500	1500	2500	3000	3500	1000	3000	1500	2500	3500	1500	2500

material *m*	revenue ($/mu)	safety stock target (mu)	cost of demand violation ($/mu)	cost of safety stock violation ($/mu)
5	1.4	1500	1	0.2
10	1	1000	1	0.2
14	1.2	1000	1	0.2

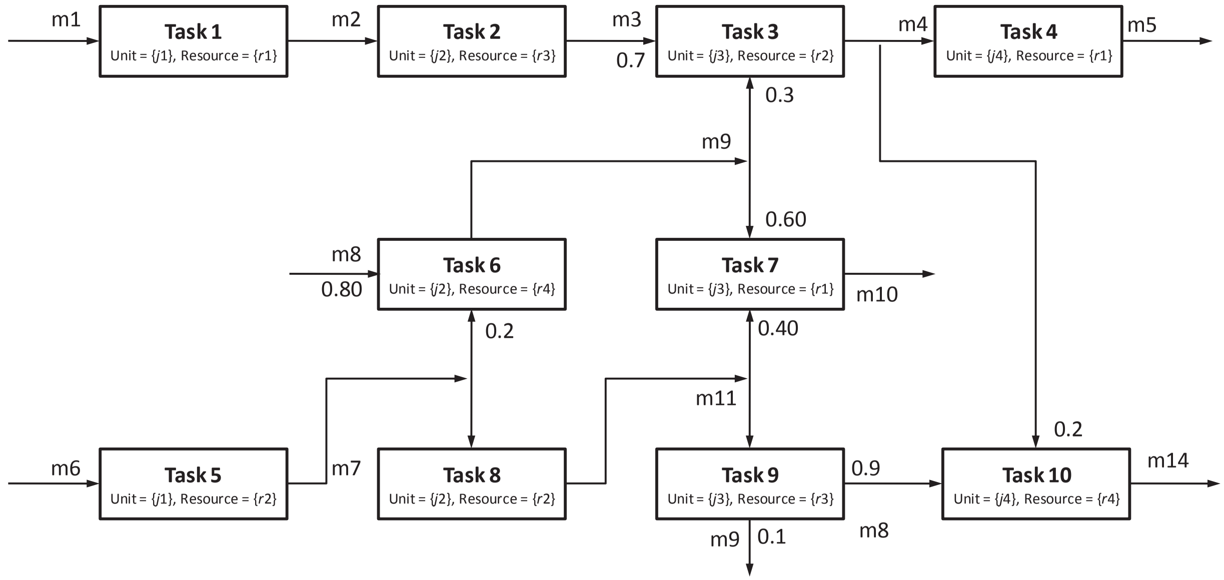

Figure 6.10 Recipe diagram for Example 2. r1 represents LP-steam, r2 represents HP-steam, r3 represents cooling water, r4 represents hot air.

An objective of $155775 was obtained. Figures 6.8 shows the Gantt chart and Figure 6.9 show product inventories.

Similarly, several scenarios based on the needs of different stake holders may be studied before a decision is made by the management.

6.4.2 Example 2

This example highlights the ability of our model in handling big problems. In this example, we consider a production facility F with 10 tasks ($i1$-$i10$), 4 units ($j1$-$j4$), 14 materials ($s1$-$s14$), and 4 resources (LP-steam, HP-steam, cooling water, and hot air). $j1$ can perform $i1$ and $i5$, $j2$ can perform $i2$, $i6$, and $i8$, $j3$ can perform $i3$, $i7$, and $i9$, and $j4$ can perform $i4$, $i10$. $i1$, $i4$ and $i7$ require LP-steam, $i3$, $i5$, and $i8$ require HP-steam, $i2$ and $i9$ require cooling water, and $i6$ and $i10$ require hot air as additional resources apart from equipment and raw materials. F produces 3 products $s5$, $s10$, and $s14$ consuming 3 raw materials $s1$, $s6$, and $s8$. Figure 6.10 gives the recipe diagram and Table 6.3 lists the data for this example. We solve this example for a planning horizon of 2 year with 24 time intervals of 30 days (720 h) each. The model statistics for all three cases are given in Table 6.4. We consider maintenance of $j1$, $j2$, $j3$, and $j4$ in $t8$, $t15$, $t2$, and $t19$, respectively. Also, we include the NPI trials in the intervals $t3$, $t6$, $t9$, $t12$, $t15$, $t18$, and $t19$.

This example consists of 3984 binary variables, 13537 continuous variables, and 24829 constraints. We considered the objective of maximising profit ($282196).

6.5 Summary

Many multiproduct batch plants employ short/long-term single-product campaigns. We addressed a routine and practical problem of campaign planning and resource allocation in such plants. Our proposed MILP model accommodates and presents novel treatment for key aspects of an industrial planning activity such as sequence-dependent changeovers, maintenance, NPIs, resource allocations, safety stock, delivery delays, etc., and gives the exact number of batches and schedule for each campaign. We successfully demonstrated the usefulness of our model for several scenarios of two moderate-size examples. The model is able to quickly optimize production planning for any given scenario, and thus has a potential to serve as a decision support tool for planners and other stake holders in practice.

7 INTEGRATED SUPPLY CHAIN PLANNING FOR MULTINATIONAL PHARMACEUTICAL ENTERPRISES[1, 2]

7.1 Introduction

The global pharmaceutical industry is grappling with tremendous turmoil in the marketplace and a dramatically changing competitive landscape. This is mainly due to the numerous mergers among different companies and the upsurge in generic manufacturers. Fierce market competition, peaking patent cliffs, mounting R&D costs, shrinking product pipelines, and stringent regulatory protocols are bringing a paradigm shift in the way pharmaceutical enterprises operate. Companies are beginning to realize that past practices will not meet future market needs. The past decade reflects a significant imbalance between new product introductions and patent losses [4]. This is expected to continue for the next few years. Also, the new products are not expected to generate the same levels of sales as the products losing patent protection. With revenue growth stalling or slowing down, companies are resorting to cost-cutting to drive bottom-line growth. Although pharmaceutical companies are not known to be the best practitioners of the supply chain models, optimization of supply chain operations is

[1] Susarla, N., & Karimi, I. A. (2012). Integrated supply chain planning for multinational pharmaceutical enterprises. Computers and Chemical Engineering, 42, 168-177.
[2] Susarla, N., & Karimi, I. A. (2011). Integrated supply chain planning for multinational pharmaceutical enterprises. Presented in ESCAPE 2011, Chalkidiki, Greece.

known to improve the bottom lines in several other industries such as airline, refining, semiconductor, etc. This has also prompted the pharmaceutical companies to begin focusing on exploiting economies of scale in manufacturing and improving the management of resources such as facilities, equipment, materials, human, information, and finances.

Figure 7.1 shows a schematic of a typical large multinational pharmaceutical enterprise. It involves functions such as raw material sourcing, primary or API (active pharmaceutical ingredient) manufacturing, secondary manufacturing, warehousing, distribution, etc. Such a configuration requires frequent transfers of materials (raw, intermediates, products, packaging, etc.) among the different sites across the globe. These material transfers not only involve time and normal operational costs, but also a slate of administrative and regulatory procedures and costs. Such costs include import duties and corporate taxes to be paid to the local governing authorities, transfer prices for material flows among the company's various sites, etc. Since the taxes and duties vary from one country to another, they can be intelligently exploited to maximize after-tax profits. Another key characteristic of a typical pharmaceutical enterprise is its high-valued material inventories. This is to ensure a high level of customer satisfaction in the face of any operational disruptions and capitalize on any unexpected opportunities (e.g. increase in demand during a disease outbreak or natural calamity). However, costly inventories freeze capital, and are undesirable for many reasons. Clearly, the pharmaceutical operations involve trade-offs, and require intelligent decision making, making operational planning and decision making a complex and crucial task.

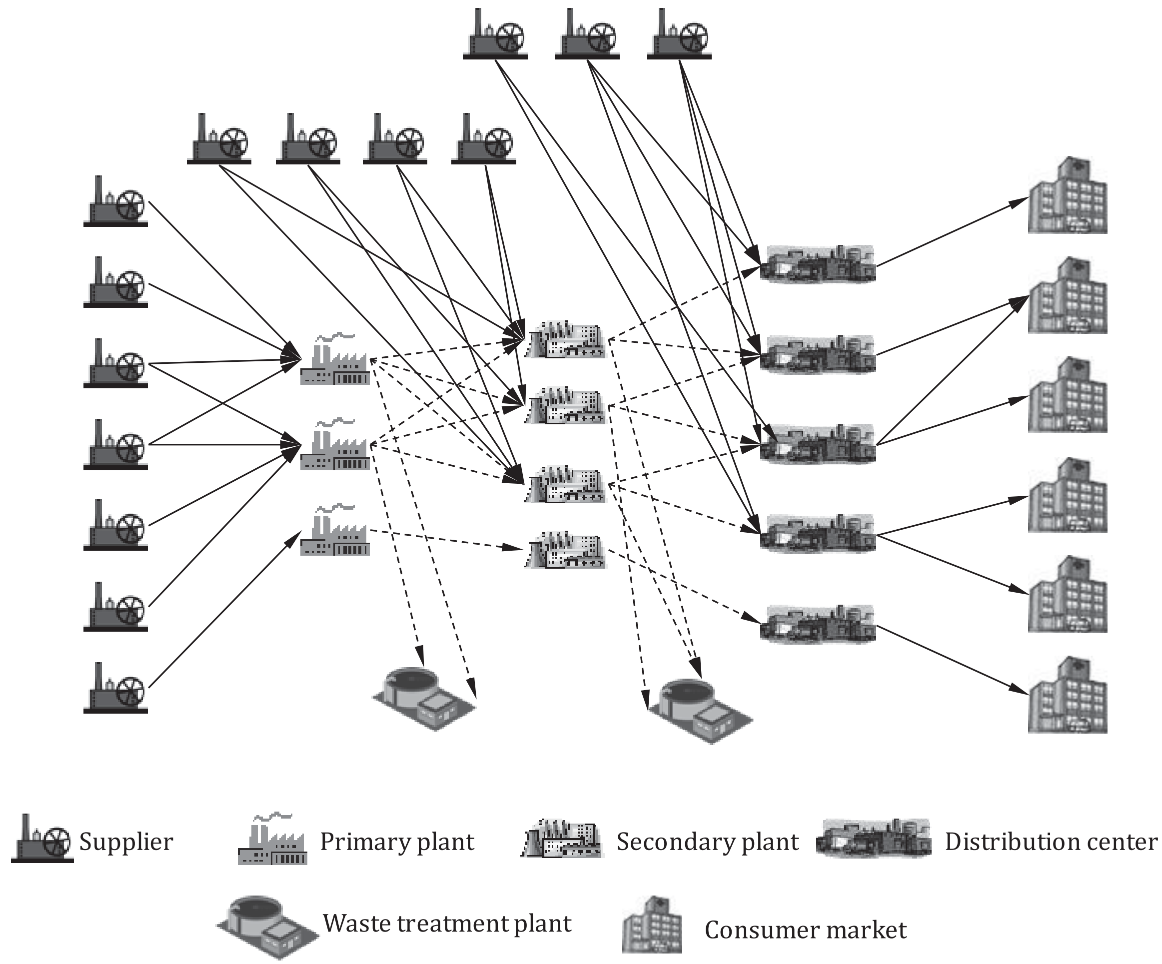

Figure 7.1 A schematic of a large multinational pharmaceutical enterprise

Now, the decisions made at the enterprise-level affect significantly the operations at individual entities (API manufacturing plants, secondary manufacturing plants, distribution houses, etc.). The entities perform many complex physicochemical transformations and value-addition steps before the drugs reach the consumer. The API plants transform raw materials into active ingredients. The secondary manufacturing plants add varieties of excipients to these active ingredients to produce drugs in their consumable forms (e.g. tablets, solutions, pastes, gels, inhalers, etc.). The distribution houses use these drugs in bulk quantities and package them in suitable sizes with appropriate labels (e.g. bottles, tablet strips and syringes) that are specifically appropriate for each market.

Most tasks described above involve multi-step batch operations that require limited and shared resources such as equipment, human, utilities, etc. A typical manufacturing (API or secondary) or packaging plant may employ several production lines to perform these operations. Figure 7.2 shows the configuration of a typical pharmaceutical plant with production lines and multi-step operations. Most plants are multipurpose batch plants that produce multiple active ingredients or products. Optimal allocation of adequate resources and sequencing of operations on production lines require involve a huge number of possible combinations, which easily becomes computationally intractable as the numbers of products and/or plants increase. In addition, pharmaceutical manufacturing is strictly and highly regulated, and operations on the same line may involve long and expensive cleaning between successive steps. Thus, holistic and integrated decision making at the enterprise level considering the nuances of individual entities and functions and their complex interactions is extremely difficult and critical for the economic sustainability of a pharmaceutical company.

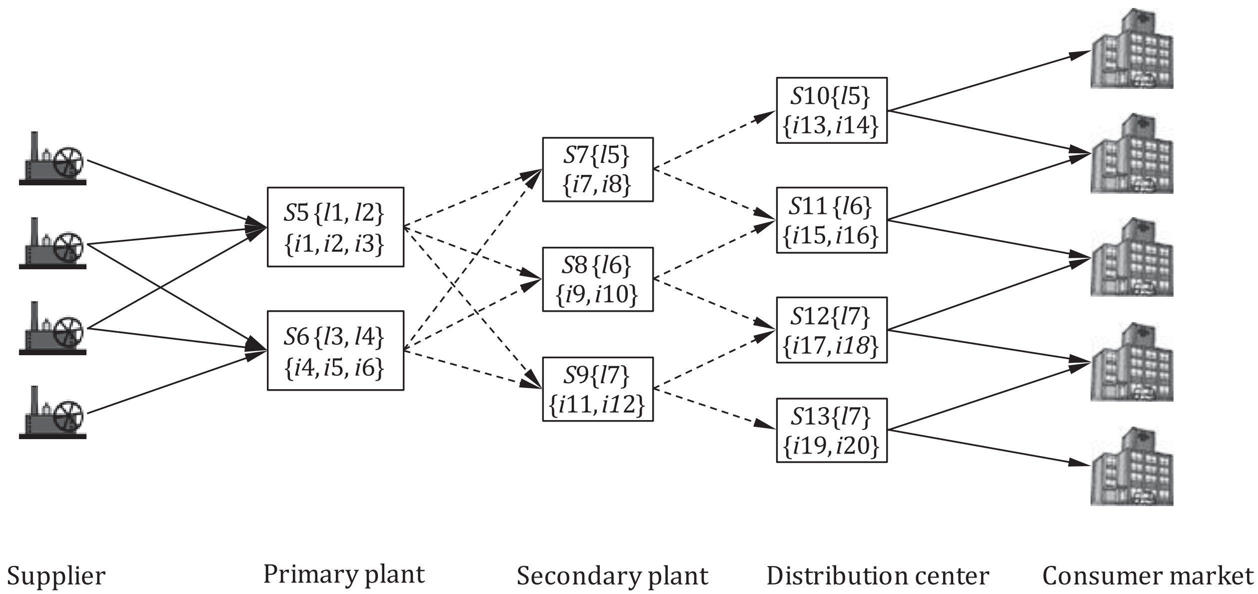

Figure 7.2 Configuration of a pharmaceutical plant with production lines and multi-stage operations

The literature review in Chapter 2 highlights that while some works have addressed the integrated problem of procurement, production, and distribution in the context of multinational pharmaceutical companies, the focus has been mainly on developing better solution strategies that improve computation time for large scale problems. However, there is a need for simpler models that are easy to implement, quick to solve, but do not compromise problem realism or features. That is the focus of this chapter. Specifically, we develop a mathematical model for the integrated problem of production planning, procurement, distribution, and inventory management in a multinational pharmaceutical enterprise, which explicitly considers the effects of regulatory affairs (including transfer prices and taxes). We consider the enterprise functions from procurement of raw materials to distribution of final products in a seamless fashion with a granularity of individual processing tasks and campaigns on production lines. Our model accommodates decisions on campaign lengths, task allocations, inventory management, shelf-lives, material transfers, transfer prices, and costs considering taxes and duties. We demonstrate the performance of our model using two case studies of multi-national companies with multiple API manufacturing facilities, several secondary manufacturing facilities, and distribution facilities located around the world.

7.2 Problem Statement

The supply chain (Figure 7.1) of a multinational pharmaceutical enterprise **E** comprises several entities, facilities, or plants. These include external raw material supplier sites, customer markets, and facilities owned by **E** such as primary/secondary manufacturing plants and distribution centres or warehouses. We view all these as globally distributed sites ($s = s1, s2, \ldots$); and define:

S = {s1, s2, s3, …} = $\mathbf{SS} \cup \mathbf{PM} \cup \mathbf{SM} \cup \mathbf{DC} \cup \mathbf{CM} \cup \mathbf{WT}$

SS = {s | site s is an external supplier of raw materials to **E**}

PM = {s | site s is a primary manufacturing site or plant within **E**, which produces APIs or *white powders*}

SM = {s | site s is a secondary manufacturing site within **E**, which produces drug formulations

DC = {s | site s is a distribution centre or warehouse within **E**, which produces drug packages}

CM = {s | site s is a consumer market for **E**}

WT = {s | site s treats waste from **E**}

Each site has several production lines ($l = l1, l2, l3, \ldots$) that perform various processing tasks ($i = i1, i2, i3, \ldots$} on various materials ($m = m1, m2, m3, \ldots$) in batches using long campaigns. We assume that each supplier site (s ∈ **SS**) has one dedicated line for each material that it sells to **E**. Similarly, every consumer site (s ∈ **CM**) has one dedicated line for each material that it receives from **E**. Waste sites have no production lines. All these lines perform only one task. Each task at any site will consume some materials (raw materials, intermediates, or packaging materials) and will produce some materials (intermediates or final products). External supplier sites (s ∈ **SS**) do not consume any material, and consumer market sites (s ∈ **CM**) do not produce any material. Define:

$\mathbf{L}_s$ = {l | production line l is at a site s ∈ **SS**}

$\mathbf{I}_l$ = {i | processing task i that production line l can perform}

M = {m | material m that **E** consumes or produces}

$\mathbf{IM}_s$ = {m | material m that site s consumes}

$\mathbf{OM}_s$ = {m | material m that site s produces}

$\mathbf{M}_s$ = {m | material m that site s consumes or produces} = $\mathbf{IM}_s \cup \mathbf{OM}_s$

We assume that each site uses long campaigns on each production line to minimize setup and cleaning during campaign changeovers. Each campaign comprises a series of consecutive and identical batches of a task. For a long campaign, we assume that the time required to process one batch of a task i can be approximated by an average processing time called batch cycle time or cycle time. In practice, production department has some flexibility in increasing/decreasing the cycle time by allocating resources appropriately. For instance, by allocating more operators, one can decrease the cycle time of a campaign. Therefore, we allow the cycle time to be determined by the allocation level of each resource. Thus, for each resource that a task needs, we have a range of resource allocation, and the cycle time varies linearly in that range. The resource with the maximum cycle time will determine the effective cycle time for the campaign.

We use recipe diagrams (Figure 7.2) [70, 206, 213] for each site to represent the details of manufacturing operations for each batch. It gives detailed information on various materials, tasks, possible production lines, etc. We define a mass ratio σ_{mi} [206] as the amount of material m consumed or produced in a single batch of a task i. $\sigma_{mi} > 0$, if task i produces material m; $\sigma_{mi} < 0$, if i consumes m; and $\sigma_{mi} = 0$, if i does not use m.

Site-to-site material transfers, with internal or external entities, are common in any multinational enterprise. However, each such transfer incurs a delivery lag or lead time. This is the time that elapses between the ordering of a material and its receipt. This lead time will clearly depend on the material and the distance between the sites. Hence, we define $\delta_{mss'}$ as the lead time for receiving a material m at site s' from site s. The geographically distributed sites also allow ample opportunities to improve its after-tax profits by carefully using materials under different tax jurisdictions. Transfer price is the price that a site in **E** charges itself for buying a material from another sister site in **E**. Although companies have some freedom to fix transfer prices, strict government regulations provide specific ranges within a given period.

Some pharmaceutical materials (especially intermediates and drugs) may be perishable. Let A_m denote the shelf-life of material m, which is the duration after its production for which the material remains suitable for use or consumption.

We now state the planning problem described in this paper as follows.

Given:

1. Sites of **E**, their types, locations, capacities, etc.
2. Products, raw materials, resources, and production recipes
3. Product demands, delivery dates, lead times, shelf lives, and planning horizon
4. Initial, safety, and maximum inventory limits
5. Import duties, taxes, and transfer price ranges
6. Costs for processing, inventory holding, material procurement, and penalties for safety stock violations

Determine:

1. Production targets for each site
2. Campaign lengths and sequences for each production line
3. Stock profiles at each site and inter-site material transfers over time
4. Transfer prices and total costs of production

Aiming for maximum after-tax profit for E.

Assuming:

1. Deterministic scenario
2. Material prices and transfer prices are piecewise constant over time

3. First in first out queuing model for inventories

4. Set-up and changeover times are lumped into campaign

7.3 MILP Formulation

Unless otherwise specified, an index takes all its legitimate values in all the expressions or constraints in our formulation. Following Karimi & McDonald [60] and Susarla & Karimi [213], we partition the planning horizon [0, H] into T = |**T**| intervals of length $h = H/T$ each, where $\mathbf{T}$ = {t = t1, t2, t3, …}. As shown in Figure 7.3, interval t begins at time $(t-1)$ and ends at time t. Also, we denote the interval just before the beginning of the horizon by t0.

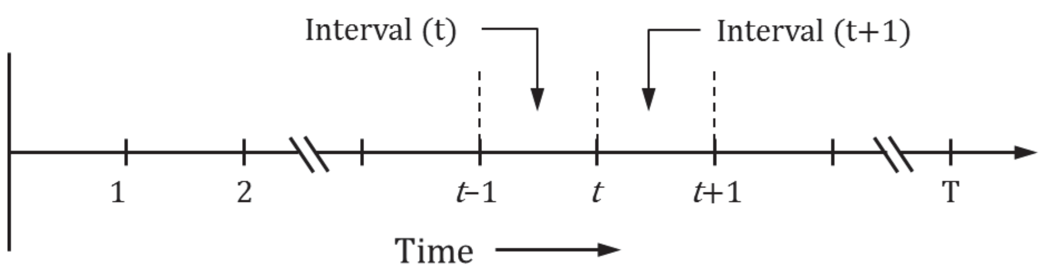

Figure 7.3 Representation of time in discrete intervals

Let n_{ilst} denote the number of batches processed in a campaign of task i on line l at site s in interval t. As discussed earlier, the cycle time for this campaign will vary with its allocated resources. Let τ_{ils}^{U} be the cycle time for minimum resource allocation, and τ_{ils}^{L} that for maximum allocation. Clearly, the shortest (longest) campaign length will be $\tau_{ils}^{L}n_{ilst}$ ($\tau_{ils}^{U}n_{ilst}$). Let cl_{ilst} denote the differential length of a campaign over and above the minimum. Then,

$$cl_{ilst} \leq \left(\tau_{ils}^{U} - \tau_{ils}^{L}\right)n_{ilst} \qquad i \in \boldsymbol{I}_l, l \in \boldsymbol{L}_s, s \in \boldsymbol{S}, t \in \boldsymbol{T} \qquad (7.1)$$

The sum of all campaign lengths on a line during an interval must not exceed the available time during that interval.

$$\sum_{i\in \mathbf{I}_l}(\tau^L_{ils} n_{ilst} + cl_{ilst}) \le h \qquad l \in \boldsymbol{L}_s, s \in \boldsymbol{S}, t \in \boldsymbol{T} \tag{7.2}$$

Now, each task will consume raw materials / intermediates and produce other intermediates / products. We assume that each site s keeps a dedicated but limited storage for each material $m \in \mathbf{M}_s$. Let Q_{mst} ($Q_{mst} \le Q^U_{ms}$) denote the net usable stock of material m after all operations (production, consumption, and inbound/outbound deliveries of all materials) during interval t have ended. Q^U_{ms} denotes the storage capacity of material m at site s. As and when needed, the site will order input materials from or ship output materials to other sites. Due to transfer delays, the material orders must be placed in advance. We assume the lead time $\delta_{mss'}$ to be an integral multiple of h, and let $OQ_{mss't}$ denote the amount of material m that site s' receives from s at time t. We assume that site s releases this material, as soon as it receives the order. Therefore, the amount of material m that site s releases at time t for delivery to s' is given by $OQ_{mss'(t+\delta_{mss'})}$. Note that we are labelling a site-to-site material transfer by its time of arrival at the destination site, rather than its time of shipment from the origin site. Then, the net stock of material m at site s at time t (or, at the end of interval t) is given by,

$$Q_{mst} = Q_{ms(t-1)} + \sum_{l\in \boldsymbol{L}_s} \sum_{i\in \boldsymbol{I}_l, \sigma_{mi}\ne 0} (\sigma_{mi} n_{ilst} B_{ils}) + \sum_{s'} OQ_{ms'st}$$

$$- \sum_{s'} OQ_{mss'(t+\delta_{mss'})}$$

$$0 \le Q_{mst} \le Q^U_{mst}, t \in \boldsymbol{T}, m \in \boldsymbol{M}_s, s \in \boldsymbol{S} \tag{7.3}$$

where, Q_{ms0} is the stock of material m at site s at time zero, and B_{ils} is batch size (the total amount of materials in one batch) of task i on line l at site s. We may not write Equation 7.3 for the sites of suppliers and consumers, if they are not a part of $\mathbf{E}$.

Equation 7.3 is valid for a material with stable physicochemical properties or no expiry date. For a material with perishable properties or a finite expiry date, the situation is more complex. We must track the age of such a material from its production to the end of its shelf life and beyond. This is to ensure that it is not used after its expiration date. For this, we label every material stock with an age and keep separate the stocks of different ages. With this, we define Q_{mst} and $OQ_{mss't}$ for a material with perishable properties as follows.

$$Q_{mst} = \sum_{a=0}^{A_m-1} Q_{mst}^{a} \qquad t \in \mathbf{T}, m \in \mathbf{M}_s, s \in \mathbf{S} \qquad (7.4a)$$

$$OQ_{mss't} = \sum_{a=0}^{A_m-1} OQ_{mss't}^{a} \qquad t \in \mathbf{T}, m \in \mathbf{M}_s, s \in \mathbf{S} \qquad (7.4b)$$

where, Q_{mst}^{a} $(a = 0, 1, 2, 3, \dots, A_m, A_m + 1)$ is the net usable stock of material m that is a intervals old at time t at site s. In other words, Q_{mst}^{a} is the amount of material that was produced during interval $(t - a)$. Likewise, $OQ_{ms'st}^{a}$ is the amount of material m with an age of a intervals received at site s from s' at time t. Note that we are labeling a site-to-site material transfer by its age at the time of arrival at the destination site, rather than its age at the time of shipment from the origin site. Thus, $OQ_{ms's(t+\delta_{ms's})}^{(a+\delta_{ms's})}$ gives the amount that site s' ships to s at time t with age a. We then use an additional variable q_{mst}^{a} $(a = 0, 1, 2, 3, \dots, A_m - 1)$ to denote the amount of material m with age a consumed by production tasks during interval t. We allow materials of different ages to mix in a batch, only if none of the materials have expired. Now, we write inventory balance for each material of a specific age to at the end of interval t at site s. We assume that a material has age 1 $(a = 1)$ at the end of the interval in which it is produced. So, we have,

$$Q_{mst}^{1} = \sum_{l \in \mathbf{L}_s} \sum_{i \in I_l, \sigma_{mi} > 0} \sigma_{mi} n_{ilst} B_{ils} - q_{mst}^{0}$$

$$m \in \mathbf{M}_s, s \in \mathbf{S}, t \in \mathbf{T} \qquad (7.5a)$$

where, q^0_{mst} denotes the amount of fresh material m consumed immediately after its production. Then, for stocks older than age 1, but not exceeding their shelf lives at time t, we write the following.

$$Q^a_{mst} = Q^{(a-1)}_{ms(t-1)} - q^{(a-1)}_{mst} + \sum_{s'} OQ^a_{ms'st} - \sum_{s'} OQ^{(a+\delta_{mss'})}_{mss'(t+\delta_{mss'})}$$

$$2 \le a \le A_m, m \in \mathbf{M}_s, s \in \mathbf{S}, t \in \mathbf{T} \qquad (7.5b)$$

Lastly, for a material that has expired, we write the following.

$$Q^{A_m}_{mst} = Q^{A_m}_{ms(t-1)} + \sum_{a \ge A_m} \sum_{s'} OQ^a_{ms'st} - \sum_{s'} OQ^{A_m}_{ms'(t+\delta_{mss'})}$$

$$m \in \mathbf{M}_s, s \in \mathbf{S}, t \in \mathbf{T} \qquad (7.5c)$$

Note that Equation 7.5c has no consumption term, as a material with age A_m or older cannot be used.

Now, a campaign of task i can consume a material m as long as the material has not expired. Therefore,

$$\sum_{a=0}^{A_m-1} q^a_{mst} = \sum_{l \in \mathbf{L}_s} \sum_{i \in \mathbf{I}_l, \sigma_{mi} < 0} |\sigma_{mi}| n_{ilst} B_{ils}$$

$$m \in \mathbf{M}_s, s \in \mathbf{S}, t \in \mathbf{T} \qquad (7.6)$$

We assume that each site has a demand $D_{mst} \ge 0$ at time t, which must be satisfied. Although it is possible to allow backlogs in our formulation, we treat the full satisfaction demand as a hard constraint.

$$\textstyle\sum_{s'} OQ_{ms'st} \ge D_{mst} \qquad m \in \mathbf{M}_s, s \in \mathbf{S}, t \in \mathbf{T} \qquad (7.7)$$

This is because demands are a high priority for the pharmaceutical industry in practice.

To respect the safety stock policy of a company, we penalize each violation of such inventory levels. In practice, the pharmaceutical industry considers two types of safety stock: overall and site-specific. The former (OSQ_{mt}) is for the total stock of a final product and all its precursor materials, which include raw materials and all the intermediates. The site-specific safety stock (SQ_{mst}) is for the stock of any material m at a given site. Usually, such overall safety stock and site-specific safety stock limits are computed by the corporate offices using complex expressions. In some companies, safety stock limits are based on the demand forecasts. Here, we assume such limits to be given. For the violations of these safety stock limits, we define ΔOSQ_{mt} for the overall safety stock of a final product, and ΔSQ_{mst} for the site-specific safety stock of a material m.

$$\Delta OSQ_{mt} \geq OSQ_{mt} - \left(\sum_{s} Q_{mst} + \sum_{m' \in \mathbf{PC}_m} \sum_{s \in \mathbf{M}_s} Q_{m'st} \right) \qquad m \in \mathbf{FP}, t \in \mathbf{T} \tag{7.8}$$

$$\Delta SQ_{mst} \geq SQ_{mst} - Q_{mst} \qquad m \in \mathbf{M}_s, s \in \mathbf{S}, t \in \mathbf{T} \tag{7.9}$$

where, **FP** is the set of final products, and $\mathbf{PC}_m$ is the set of all precursor materials (raw materials and intermediates) of the final product m.

Since the sites within **E** are geographically distributed, transfers across international tax jurisdictions are inevitable. The price at which a material is transferred among the two sister sites of **E** is called transfer price. These prices are very important for both the companies and the tax administrations, as they play a crucial role in determining the taxable incomes of the business units of **E** in different tax jurisdictions. Transfer price is usually set by the enterprise itself to take advantage of different tax regimes. However, this setting is highly governed by the tax authorities. The tax authorities provide a range of transfer price based on strict guidelines for each material. While the companies are free to choose a suitable transfer price from this range, they cannot change it every interval. It must remain constant for a period of several intervals. Let $\mathbf{P} = \{p = p1, p2, p3, ...)$ denote the set of periods within which the transfer price remains constant. A period p may span several intervals. To model this, we define $\boldsymbol{T}_p$ = $\{t \mid t$ is in period $p\}$, and treat each transfer price as an optimization variable within a specified range $[TP^L_{mss'p}, TP^U_{mss'p}]$. We also treat the raw material and final product prices as transfer prices with no loss of generality, as we can fix them by setting appropriate bounds. To compute the cost of procuring input materials, let $\Delta TP_{mss'p}$ by the following:

$$TP^L_{mss'p} \sum_{t \in \mathbf{T}_p} OQ_{mss't} + \Delta TP_{mss'p} \leq TP^U_{mss'p} \sum_{t \in \mathbf{T}_p} OQ_{mss't}$$

$$m \in \mathbf{M}_s, s, s' \in \mathbf{S}, p \in \mathbf{P} \qquad (7.9)$$

Pharmaceutical manufacturing produces a variety of wastes (e.g. solvent, water, expired materials, volatile organic compounds or VOC, etc.) that are treated or disposed in several ways including incineration, wastewater treatment, etc. The treatment and disposal of wastes incur cost. In this formulation, we do not differentiate between waste treatment and waste disposal sites. Instead, we combine them and call such sites as waste handling sites. This is because we do not explicitly model waste treatment process. We force waste materials and expired materials to move to appropriate treatment sites by setting $Q^U_{mst} = 0$ for such materials. We also set appropriate Q^U_{msp} to reflect the capacity of such a waste handling site in a period p.

To maximize the utilization of plant resources such as production lines, we define PU_{lsp} as the total idle time of production line l at site s during period p:

$$PU_{lsp} = |\mathbf{T}_p|h - \sum_{t \in \mathbf{T}_p} \sum_{i \in \mathbf{I}_l} (\tau^L_{ils} n_{ils} + cl_{ilst}) \qquad l \in \mathbf{L}_s, s \in \mathbf{S}, p \in \mathbf{P} \qquad (7.10)$$

The total operating cost for a site s during tax period p is given by,

$Cost_{sp}$ = Processing + Procurement + Inventory holding +

Safety stock penalties + Import duties + Waste handling (i.e. treatment/disposal)

+ Penalty for plant idling (7.11)

Processing: $\sum_{t\in\mathbf{T}_p}\sum_{l\in\boldsymbol{L}_s}\sum_{i\in\boldsymbol{I}_l}\left\{a_{ils}-b_{ils}\left(\frac{\frac{\tau_{ils}^{L}n_{ilst}+cl_{ilst}}{n_{ilst}}-\tau_{ils}^{L}}{\tau_{ils}^{U}-\tau_{ils}^{L}}\right)\right\}n_{ilst}$

where, a_{ils} and b_{ils} are suitable constants. The above assumes processing costs to vary linearly with the average cycle time. Simplifying the above for period p at site s gives,

Processing: $\sum_{t\in\mathbf{T}_p}\sum_{l\in\boldsymbol{L}_s}\sum_{i\in\boldsymbol{I}_l}\left[a_{ils}n_{ilst}-\left(\frac{b_{ils}cl_{ilst}}{\tau_{ils}^{U}-\tau_{ils}^{L}}\right)\right]$

Procurement: $\sum_{s'}\sum_{m\in\mathbf{IM}_s}(TP_{ms'sp}^{L}\sum_{t\in\mathbf{T}_p}OQ_{ms'st}+\Delta TP_{ms'sp})$

Inventory holding: $\sum_{t\in\mathbf{T}_p}\sum_{m\in\mathbf{M}_s}hc_{msp}Q_{mst}$

Safety stock penalties: $\sum_{t\in\mathbf{T}_p}\sum_{m\in\mathbf{M}_s}\gamma_{msp}\Delta SQ_{mst}$

Import duties: $\sum_{s'}\sum_{m\in\mathbf{IM}_s\cap\mathbf{OM}_{s'}}d_{ms'sp}(TP_{ms'sp}^{L}\sum_{t\in\mathbf{T}_p}OQ_{ms'st}+\Delta TP_{ms'sp})$

Waste handling:

$$\sum_{s'\in\mathbf{WT}}\sum_{m\in\mathbf{IM}_{s'}}\left(TP_{mss'p}^{L}\sum_{t\in\mathbf{T}_p}OQ_{mss't}^{A_m}+\Delta TP_{mss'p}\right)+$$

$$\sum_{s'\in\mathbf{WT}}\sum_{m\in\mathbf{IM}_{s'}}(TP_{ms'sp}^{L}\sum_{t\in\mathbf{T}_p}OQ_{ms'st}+\Delta TP_{ms'sp})$$

Plant idling penalty: $\sum_{p\in\mathbf{P}}\sum_{s\in\mathbf{S}}\sum_{l\in\boldsymbol{L}_s}CI_{lsp}PU_{lsp}$

where, hc_{msp} is the unit cost for holding a material m at site s for period p, γ_{msp} is the safety stock penalty per unit amount, $d_{ms'sp}$ is the unit import duty, and CI_{lsp} is the cost of idling during period p for line l at site s. Note that we have two terms for waste handling, where the first is for the expired materials and the second is for the wastes.

The revenue (R_{sp}), taxable income (IBT_{sp}), and after-tax profits (ATP_{sp}) for site s considering a depreciation rate of Dep_{sp} and a tax rate of tax_{sp} at site s for period p are given by,

$$R_{sp}=\sum_{s'}\sum_{m\in\mathbf{IM}_{s'}}\left(TP_{mss'p}^{L}\sum_{t\in\mathbf{T}_p}OQ_{mss't}+\Delta TP_{mss'p}\right)\tag{7.12}$$

$$IBT_{sp} \geq R_{sp} - Cost_{sp} - Dep_{sp} \tag{7.13}$$

$$ATP_{sp} = R_{sp} - Cost_{sp} - Dep_{sp} - tax_{sp}IBT_{sp} \tag{7.14}$$

Then, the total profit (NP) for **E** is the objective for our planning model.

$$\text{Maximize } NP = \sum_p \sum_s ATP_{sp} \tag{7.15}$$

While Equations 7.15 does not discount cash flows, accounting for time value of money is straightforward to do. Furthermore, we can easily deal with scheduled NCEs (new chemical entities) testing and maintenance in our formulation. For this, we define these as additional tasks with only one batch per campaign and cycle times as their durations. As indicated by Susarla and Karimi [213], plant managers in a typical batch plant often tweak resource allocations to campaigns to increase/decrease productivity. In this work, we have assumed that the resource allocation profiles are available, and have captured their effect in the upper limit ($n_{ilst} \leq n_{ilst}^{U}$) on batches in a campaign during interval t. In practice, the plant personnel have good estimates of n_{ilst} based on experience.

This completes our operational planning model (SK-1, Equations 7.1–7.15) for a global pharmaceutical enterprise.

7.4 Solution Algorithm and Numerical Evaluation

While getting a very quick solution is not necessarily critical for industry-scale long-term (e. g. 5-10 years) planning, computational tractability is obviously important. As discussed previously, much research has focused on solution strategies such as hierarchical modelling and mathematical decomposition. While we have kept our model largely linear, it has integer variables (n_{ilst}). It can be intractable for large enterprises and long horizons (e. g. Example 2). Therefore, we used the following heuristic strategy.

For most large problems, we expect many n_{ilst} to be zero in a solution of the relaxed MILP. To reduce MILP size, we assume that these variables will remain zero in the optimal solution. This strategy enabled us to at least solve our model for large problems with little compromise on solution quality.

To test our algorithm and demonstrate our model, we consider two examples that are based on the operations of multinational pharmaceutical companies. For our evaluation, we use CPLEX 12 (with default options)/GAMS 23.7 on a Dell Precision T5500 workstation with Intel® Xeon® 2 x 2 GHz CPUs, 4 GB RAM, running Windows 7 Professional® 64-bit operating system.

7.4.1 Example 1

A pharmaceutical enterprise has 18 sites ($s1$–$s18$) including four raw material suppliers ($s1$-$s4$), two API or primary manufacturing plants ($s5$-$s6$), three secondary manufacturing plants ($s7$-$s9$), four packaging & distribution centres ($s10$-$s13$), and five consumer markets ($s14$-$s18$). Its operations involve 24 materials ($m1$–$m24$: 4 raw materials, 12 intermediates, and 8 final products), 20 production tasks ($i1$–$i20$: 6 in primary plants, 6 in secondary plants, and 8 in packaging/distribution houses), and 11 production lines ($l1$–$l11$). Figure 7.4 shows a schematic of **E**.

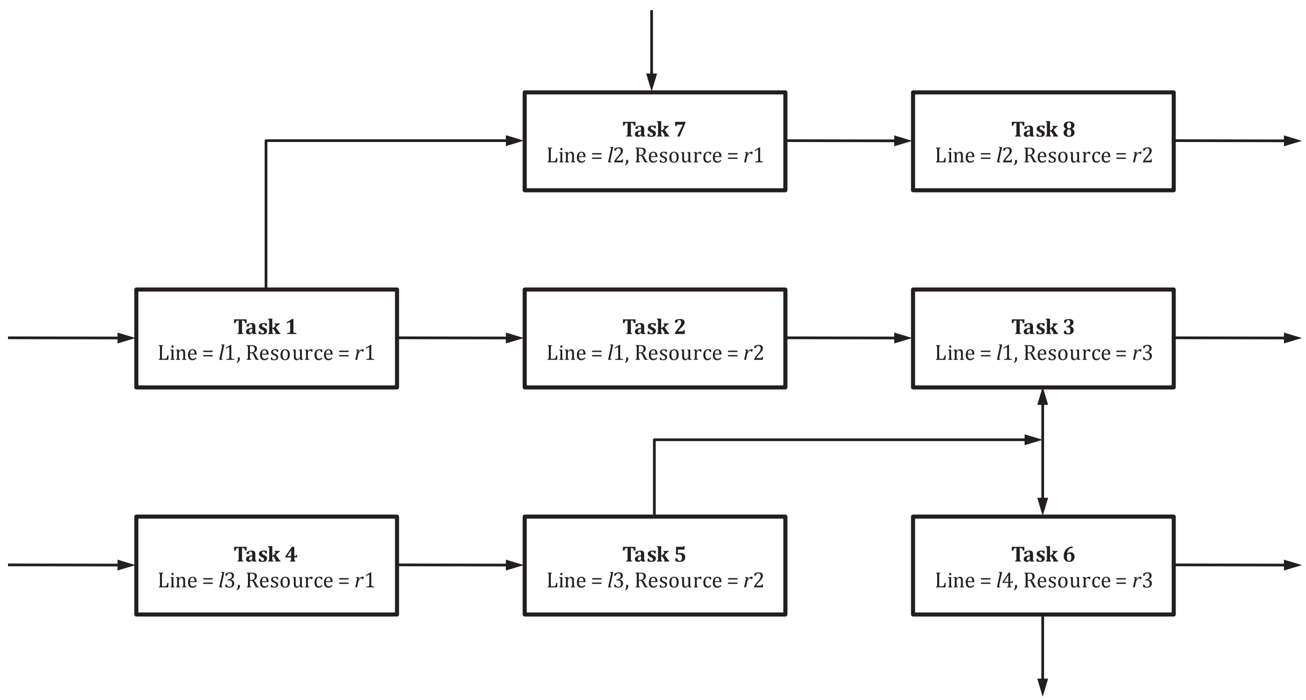

Figure 7.4 Schematic of E for Example 1

Sites $s5$ and $s6$ perform ($i1$, $i2$, $i3$) on ($l1$, $l2$) and ($i4$, $i5$, $i6$) on ($l3$, $l4$). $s7$, $s8$, and $s9$ perform ($i7$, $i8$) on $l5$, ($i9$, $i10$) on $l6$, and ($i11$, $i12$) on $l7$. $s10$, $s11$, $s12$, and $s13$ perform ($i13$, $i14$) on $l8$, ($i15$, $i16$) on $l9$, ($i17$, $i18$) on $l10$, and ($i19$, $i20$) on $l11$. We assume that all production lines are free at time zero and all materials have shelf lives longer than the planning horizon of 6 months. We assume that the planning horizon has only one period and comprises 8 time intervals of 540 h each. We also assume a lead time of $\delta_{mss'} = 1$ for all inter-site material transfers. Only the initial inventories of raw materials are available and those too at primary sites only. Furthermore, only demands are for final products and those too at the end of the horizon only.

Being a relatively small problem, we solve it as an MILP to get an objective of \$531,896 with a gap of 0.39% in about 2000 CPUs. Our heuristic strategy of fixing some integers to zero also gives us a close objective of \$531,885 in around 500 CPU s. For both cases, we allowed up to 2000 CPU s, but the objective did not improve much. This example consists of 1168 constraints, 1573 continuous variables, and 182 discrete variables. The rMIP solution gave 100 integer variables at zero and an objective of \$535,265. In the next step, we fixed these 100 variables to zero, and then solved the original MILP problem with only 82 integer variables. Table 7.1 lists the model statistics for both (full-scale MILP and LP-MILP) solution methods. This example, although relatively small, is rich with relevant features and verifies our heuristic strategy. Note that the solution obtained from our heuristic approach is within 0.7% of the rMIP solution. Thus, we can claim it to be a good solution. While the reduction in computation time from our heuristic approach is not evident in this example, it will be apparent for large problems such as Example 2 discussed next.

Table 7.1 Model statistics for Example 1

Model Statistics	MILP	LP - MILP
Equations	1168	
Non-Zeros	4346	
Continuous Variables	1573	
Discrete Variables	182	82
Objective	531896	531885

7.4.2 Example 2

An enterprise (Figure 7.1) has 34 sites including 14 suppliers (for raw, excipient, and packaging materials), 3 primary manufacturing plants, 4 secondary manufacturing sites, 5 packaging and distribution centres, 6 consumer markets, and 2 waste treatment / disposal sites. It involves 62 materials (14 raw materials including excipients and packaging materials, 35 intermediates, 10 final products, and 3 wastes), 45 production tasks, and 34 production lines. Thus, this is a larger and more complex example than Example 1.

Figures 7.5 and 7.6 show the multi-stage production recipes at the primary and the secondary plants, respectively. We assume a lead time of $\delta_{mss'} = 1$ for all inter-site material transfers. All production lines are free at time zero. Four materials have a shelf life of seven periods ($A_m = 7$). Primary manufacturing plants purchase raw materials and maintain corresponding stocks, secondary plants consume/hold excipients, and distribution centres use/hold packaging materials. However, only raw materials have initial inventories. Several orders have been placed by the sites before time zero. No storage is available for wastes, so they are sent immediately for treatment/disposal. The planning horizon is 60 months (5 years) with 20 periods of a quarter each and 60 time intervals of 720 h each. Final product demands are known at the end of each quarter.

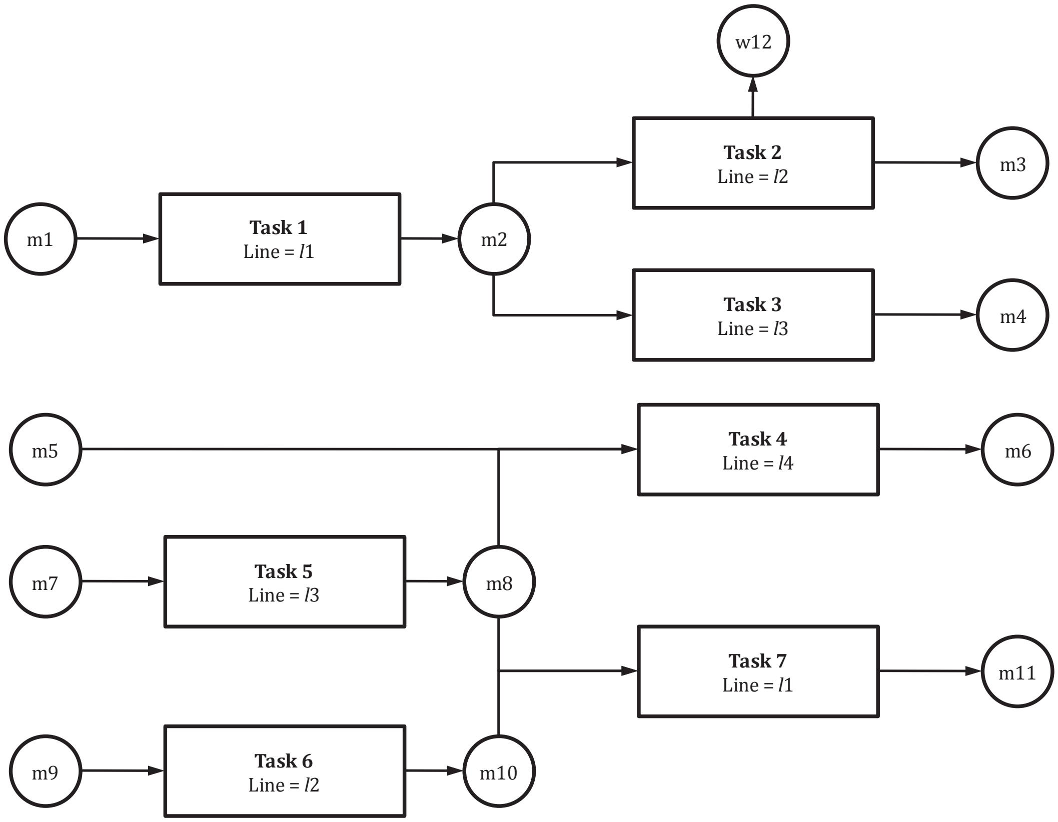

Figure 7.5 Multi-stage configurations of primary plants in Example 2

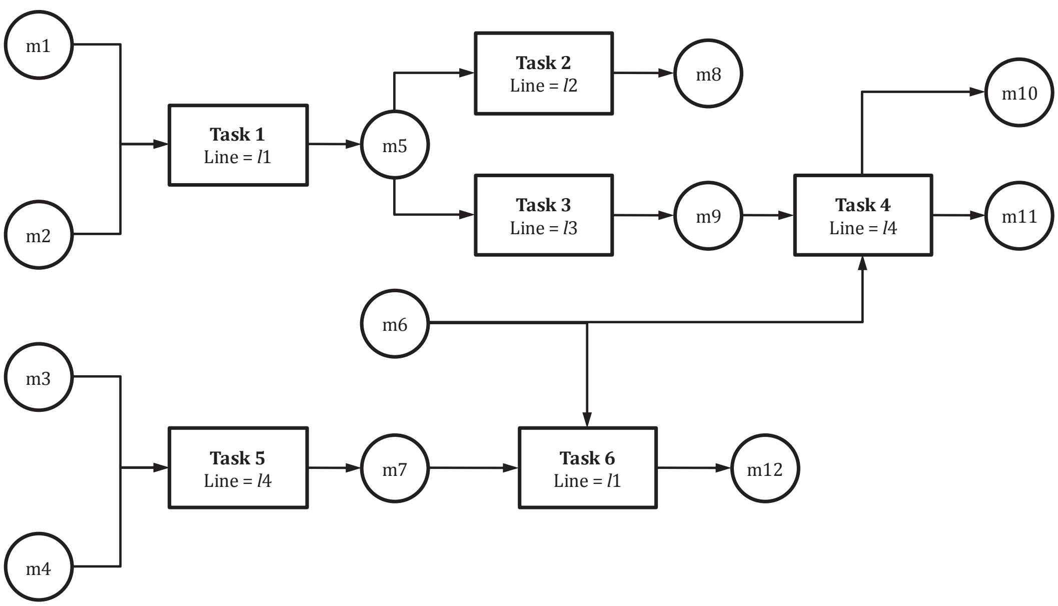

Figure 7.6 Multi-stage configurations of secondary plants in Example 2

This large example involves 35,431 constraints, 79,698 continuous variables and 3,190 integer variables. We first try to solve this example as an MILP. No solution is obtained even after 5 h of CPU time. Then, we use our heuristic strategy. The rMIP gives an objective of $360,279. We identify the integer variables at zero and fix them. We then solve the original MILP with fewer integer variables and get a solution of $355,583 with a gap of 1.08% in 300 CPU s. This again is an acceptable solution in practice and is within 1.3% of the rMIP objective. The reduced MILP obtained a feasible solution ($351,330) within the first 17 CPU s. Our approach solves this large problem with acceptable quality and in reasonable computation time. Table 7.2 lists the model statistics for both methods.

Table 7.2 Model statistics for Example 2

Model Statistics	MILP	LP - MILP
Equations	35431	
Non-Zeros	139471	
Continuous Variables	79986	
Discrete Variables	3190	187
Objective	na*	355583

* No solution was achieved within 5 hrs CPU time

7.5 Summary

While global integrated enterprise-wide planning has attracted some academic interest with some work on pharmaceutical industry, its focus has largely been on efficient solution strategies for large problems. We have presented a simple MILP model for multi-period enterprise-wide planning in a multi-site, multi-echelon, and global network of a pharmaceutical company. One key aspect of our model is to consider the entire enterprise in a seamless fashion with a granularity of individual task campaigns on each production line. The model integrates procurement, production, and distribution along with the effects of international tax differentials, inventory holding costs, material shelf-lives, waste treatment / disposal, and other real-life factors on the after-tax profit of the company. In addition, our proposed LP-MILP algorithm seems to work well on two examples based on an existing pharmaceutical company. Thus, this work has the potential to serve as a decision-support tool for long-term planning.

8 PLANPERFECT: A DECISION-SUPPORT TOOL FOR PHARMACEUTICAL PRODUCTION PLANNING AND RESOURCE ALLOCATION[1, 2]

8.1 Introduction

Ever-changing market dynamics, fluctuating demands, stringent regulatory protocols, volatile energy prices, shrinking product pipelines, and peaking patent cliffs are posing unprecedented challenges on the economic sustainability of global pharmaceutical companies. To remain competitive and economically sustainable, companies now increasingly seek new and innovative technologies to reduce costs and improve profit margins. There are a number of areas, where a pharmaceutical company can reduce cost such as product development, stock policies, manufacturing, etc. In this regard, several studies [7, 8, 214] estimate the possible annual savings in the pharmaceutical manufacturing to be in the range of $20 – 50 billion. Thus, there is an increasing need and interest of pharmaceutical companies to optimize manufacturing operations by reducing operational costs and conserving resources. The complex and combinatorial nature of its operations in which many products and intermediates share plant

[1] Susarla, N., & Karimi, I. A. (2012). Intelligent decision-support tools for effective and integrated operational planning in pharmaceutical plants. Presented in PSE2012, Singapore, July 15 – 19.

[2] Susarla, N., & Karimi, I. A. (2012). PlanPerfect: An integrated production planning and decision-support tool for pharmaceutical plants. Presented in ESCAPE 2012, London, UK, June 17 – 20.

equipment and resources in a dynamic manner makes production planning the most vital component to this endeavor.

Now, the manufacturing process in a pharmaceutical company is inherently complex and involves multiple manufacturing stages (active pharmaceutical ingredients and drug formulation) (Figure 8.1). Each of these stages require multiple raw materials and a number of other resources (e.g. manpower, utilities, electricity, and equipment), to produce different intermediates and products. However, such resources are expensive and limited. Thus, an efficient manufacturing process requires an effective allocation and usage of resources to meet production targets. For this, the management regularly plans resource allocations and schedules process operations in the best interest of the company. Planning in such companies involve myriad decisions, usually, for a period of 2 or 3 years. This primarily involves determining campaign lengths and their sequences, resource allocation, inventory management, maintenance plan, and new product introductions (NPIs). Also, as mentioned in Chapter 6, the planning activity demands collaboration with different departments (process, maintenance, laboratory, sales, and suppliers) for various inputs (demands, resource availability, and maintenance). Furthermore, given that most of the decisions and the related inputs are either estimates or forecasts, the values for the inputs keep changing with time. This requires frequent changes or modifications to the existing production plan.

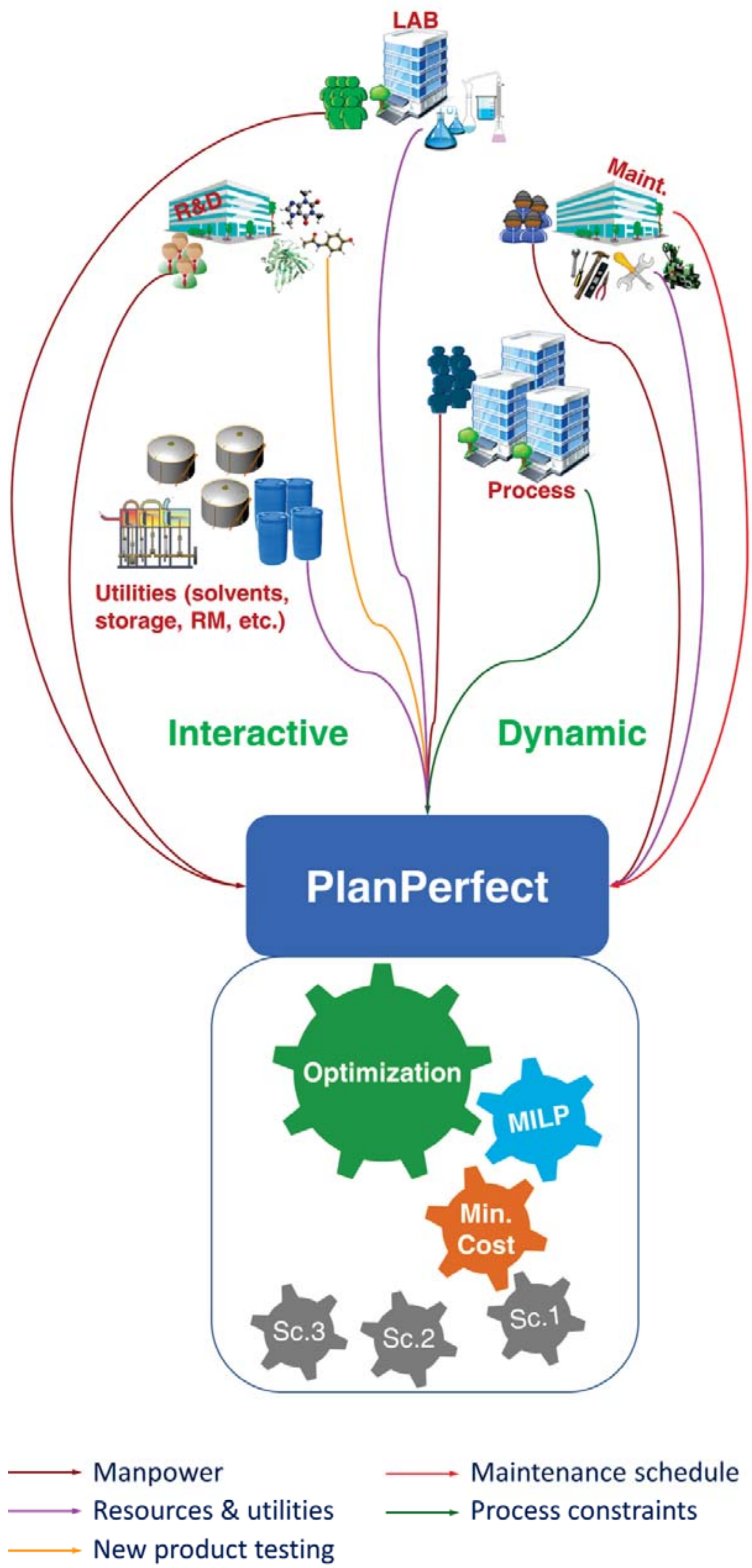

Figure 8.1 Integrated and complex pharmaceutical planning.

Unique characteristics of pharmaceutical plants and the criticality of effective decision making demand sophisticated decision-support tools for efficient resource allocation and production planning to achieve an optimal plan. The existing commercial software packages (e. g. Oracle ERP, SAP ERP, etc.) are limited by scope and are primarily designed to extract transactional efficiency rather than intelligent and optimized decisions that impact the bottom line in much more profound ways. These modular packages consist of planning modules such as – advanced planning and scheduling (APS), manufacture resource planning (MRP 2), etc. These modules not only tend to be too generic to satisfy the specific needs and constraints of the pharmaceutical plants but also fall short in evaluating different scenarios and their effect on the overall profit of the plant. In addition, such commercially available tools are difficult to use and require special trainings. They offer less freedom to the planners in altering process parameters or plant configurations. As a result of this, planners in most companies resort to either making ad hoc decisions or using simple spreadsheets for planning. These approaches are time consuming (usually, 2 to 3 days for a scenario) and often yield solutions with ample room for improvement.

In this chapter, we present a blueprint of a smart production planning and resource allocation tool PlanPerfect, which addresses the specific needs and constraints of planners and other stakeholders in a pharmaceutical plant. It consists of a customized GUI and provides quick solutions to the production planning problem. Importantly, the tool runs an optimization model that is specifically designed and tailored to the needs and constraints of pharmaceutical companies. The tool embodies a master database consisting of relevant details on plant-specific process configurations and an optimization model consisting of a variety of operational constraints. For a given set of parameters and a planning scenario, the model determines an operational strategy (i.e., plan) such that a chosen objective (min cost, inventory, wastes, etc.) is optimized. Next, we briefly describe the specific problem we address in this chapter and then, we present the basic architecture and framework of PlanPerfect. Finally, we list a few important features of PlanPerfect and illustrate them using an industrial-scale example.

8.2 Problem Statement

The problem we address in this chapter consists of two sub-parts. First is to develop a customized and user-friendly GUI. Second is to develop a model for production planning and resource allocation. For the first, a few of the important considerations for designing the GUI are the simplicity in implementation, ease of use, and requirement of no additional skills. The tool has a number of stake holders ranging from planners and managers (maintenance, process, lab, and R&D) to higher management and logistics partners. Every stake holder has a specific need, which translates into the desired features of the GUI. Importantly, the tool should be able to adapt itself to the existing format of planning data (if there exists any). Also, a comprehensive, interactive, and aesthetic dashboard is of high importance. All scenario specific data and results need to be in a desired format and be easily accessible (i.e. with lesser manual effort). Finally, an interactive error detecting, tracking, diagnosing, and mitigating system is essential to make the tool immune to human errors. This will make the tool more robust and easier to use for a varied range of users (i.e. from operators to higher management).

For the second, we consider a production planning and resource allocation for a pharmaceutical manufacturing facility (F), which houses T batch production trains (t = 1, 2, ..., T) to produce P APIs (p = 1, 2, ..., P) and IP intermediates (ip = 1, 2, ..., IP).

The production process involves *PI* (*pi* = 1, 2, ..., *PI*) production stages and *S* different materials, which include raw materials (*RM*), Intermediate materials (*IP*), APIs (*P*), and Wastes (*W*), i.e. $S = RM + IP + P + W$. In adherence to the 'Regulatory Protocols' by the 'Food and Drug Administration of the United States (FDA), the production of each stage *i* is 'registered' with one or more of the trains *t*. However, a train t can process multiple stages. In addition, at any time a train *t* may either be idle (*i*0), under maintenance (*im*), and processing either registered products (*i*) or New Chemical Entities (NCEs) (*in*). To describe the suitability of production stages with the processing trains, we define $I_t = \{i \mid \text{train } t \text{ processes stage } i \in \mathbf{I}\}$, where $\mathbf{I} = PI + \{i0, im, in\}$.

As described in Chapter 6, each production train *t*in *F* consists of a series of equipment such as mixer, reactor, crystallizer, and drier. Each production stage is processed in terms of individual batches. For each batch of stage *i* on train *t*, we associate a constant batch size B_{it} and processing time P_{it}. While P_{it} refers to the total time for a batch to pass through all the individual equipment within a train *t*, a batch has a smaller residence time in each of the individual equipment. Therefore, the cycle time (C_{it}), which is defined as the time interval between feeding two consecutive batches, is often much shorter than the processing time (P_{it}). The cycle time is pre-computed during the plant-fit of the process by the technical department based on all residence times within a train and their equipment cleaning times, which is much before the planning. Thus, planners treat cycle time (C_{it}) as a given parameter.

The turn-around time (inter-product or inter-stage cleaning time and set-up time) is usually quite long and is both cost and manpower intensive. Thus, to avoid frequent changeovers, the plant is operated in campaigns of several batches. Each batch of a stage consumes some materials (raw materials or intermediates and other additives, if any) and produces other materials (intermediates, products, and/or wastes). We group the materials consumed and produced by a production stage into two different sets ISC_s = {i | material s is consumed by stage i} and ISP_s (= {i | material s is produced by stage i}), respectively. While, in principle, mass is always conserved across all equipment, industrial operation involve losses during the cleaning or separation. To eliminate the inconsistency of balance due to losses, we define yield (γ_{it}) as the ratio of the actual quantity of product or wastes recovered to the batch size. Theoretically, yield (γ_{it}) should be always less than 1. However, as we do not define the various product specific additives as a different material resource (additives), the yield sometimes may even be more than 1.

The storage of intermediates may involve various storage capacities and wait policies. These are unlimited intermediate storage (UIS), limited intermediate storage (LIS), no intermediate storage (NIS), unlimited wait (UW), limited wait (LW), and zero wait (ZW). Each stage on a processing train begins (ends) with the transfers of input (output) materials into (out of) that unit from (to) appropriate storage facilities. To deal with the uncertainty in the demand forecasts of products, the product stock is always maintained over and above a safety limit (Inv^{Safe}). Also, the plant may have pre-defined occupancy for maintenance (*im*) or processing of NCEs (*in*) for clinical trials, which may constrain the production process.

8.3 PlanPerfect Framework

In the following, we restrict ourselves to the discussion on basic underlying concept and key integrated components of PlanPerfect.

8.3.1 Concept

PlanPerfect is built on a novel construct of synergistically using planner's knowledge base along with the mathematical programming. One of the key ideas here is that the availability and the allocation of limited resources directly affects the productivity of a plant. For instance, pharmaceutical plant operations are human intense and so, a limited availability of operators directly affects the number of batches that can be processed in a week. Also, under the limited availability of operators, production of some products can be expedited by allocating sufficient operators at the cost of impeding the production of other products. Clearly, without the consideration of resource allocation and availability, a given plan may either underestimate or overestimate the total production. Typically, plant managers deal with limited resources by making ad-hoc and in-prompt decisions. The plant is then operated in a mode different from normal. A mode of operation can then be defined as the plant operating procedure under the limitation of one more resources with a specific resource allocation and usage profile. There can be m (m = 1, 2, …, M) such modes, e.g. low/high throughput mode, vacation mode, lab constrained mode, etc. In the normal mode of operation all resources are assumed to be available in sufficient quantities. Every mode of operation has its own resource allocation profile. Thus, each mode differs from the other because the plant's productivity (number of batches processed in a week or a month) is a function of its mode of operation. Clearly, planning models must consider this variation in productivity for generating realistic targets.

$$\textit{Plant productivity} = f(\textit{mode of production}, m) \qquad \text{(a)}$$

In principle, the availability and allocation of resources can be modelled using a detailed scheduling model. However, in most of the scenarios, the resource availability data is not known a priori. Thus, a detailed scheduling or stochastic planning may make the complete model too complex to solve and hence, hinder its industrial usage. Another way to capture the variation in productivity is to include the knowledge and experience of the planners in modelling productivity for different modes of operation. In addition, stake holders of a plan (planners, process engineers, laboratory officials, maintenance engineers and technicians, etc.) provide important insights of the entire process. Figure 8.2 represents the framework used to construct PlanPerfect.

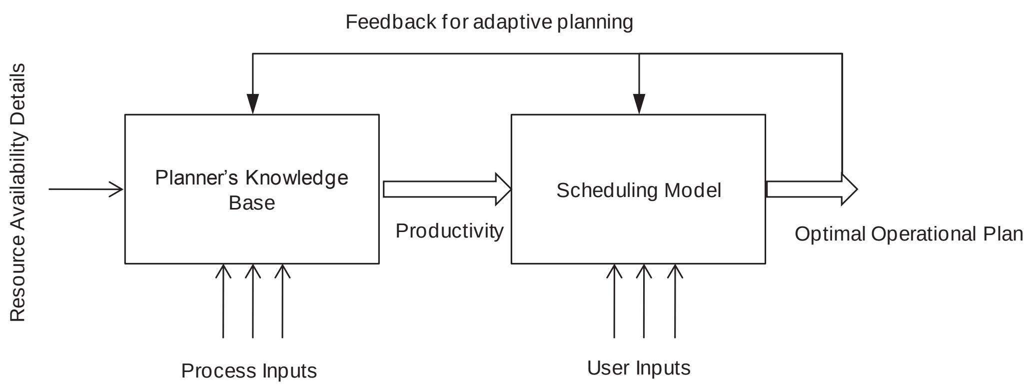

Figure 8.2 Framework used to construct PlanPerfect

Another important aspect is an effective adaptive planning framework to accommodate any change in a planning scenario (with respect to resource availability or user inputs), as soon as a new information is available. Thus, a feedback mechanism that evaluates the current plan against the revised scenario and communicates useful information back to the scheduling model is critical. For PlanPerfect, the feedback is either to the scheduling model or to the planner's knowledge base. This helps in making realistic changes to the existing plans. It further gives important insights on the impact of the revised plan on resource allocation (if any) and thus on the productivity.

8.3.2 Components

The tool consists of a GUI based on Microsoft Excel, where the user can give all related inputs and do appropriate changes to model a specific scenario. The tool then uses CPLEX 12 solver on GAMS 23.2 platform to optimize the plan. The optimal plan is then displayed on the output interface, which is again based on MS-Excel. The tool consists of a number of algorithms and functions for pre-processing of user-input form to the desired form. The inputs are transferred to the optimization module and retrieves GAMS output into the output interface after processing and consolidating results into a desired form. The in-built functions help to convert the results from GAMS into a graphical format preferred by the user. Figure 8.3 shows the schematic of the components in PlanPerfect. All these functions are developed using Visual Basic in MS Excel. Inputs and outputs are grouped into specific clusters and are presented in different 'worksheets' of the same MS-Excel 'workbook'. Each such input or output is readily accessible from the main worksheet, also known as dashboard. Furthermore, for the purpose of scenario evaluation and future reference, the user can use the export function in the tool to save all the related inputs and outputs of a scenario into a new workbook at a desired location. Also, a user can re-load all the related information with respect to an old planning scenario using the import function of the tool. The main solver or the optimization module (CPLEX on GAMS platform) of PlanPerfect is opaque to user and runs only in the background. To generate a desired planning scenario, the user is allowed to do all required changes using the GUI. Following are the user-defined inputs for the tool.

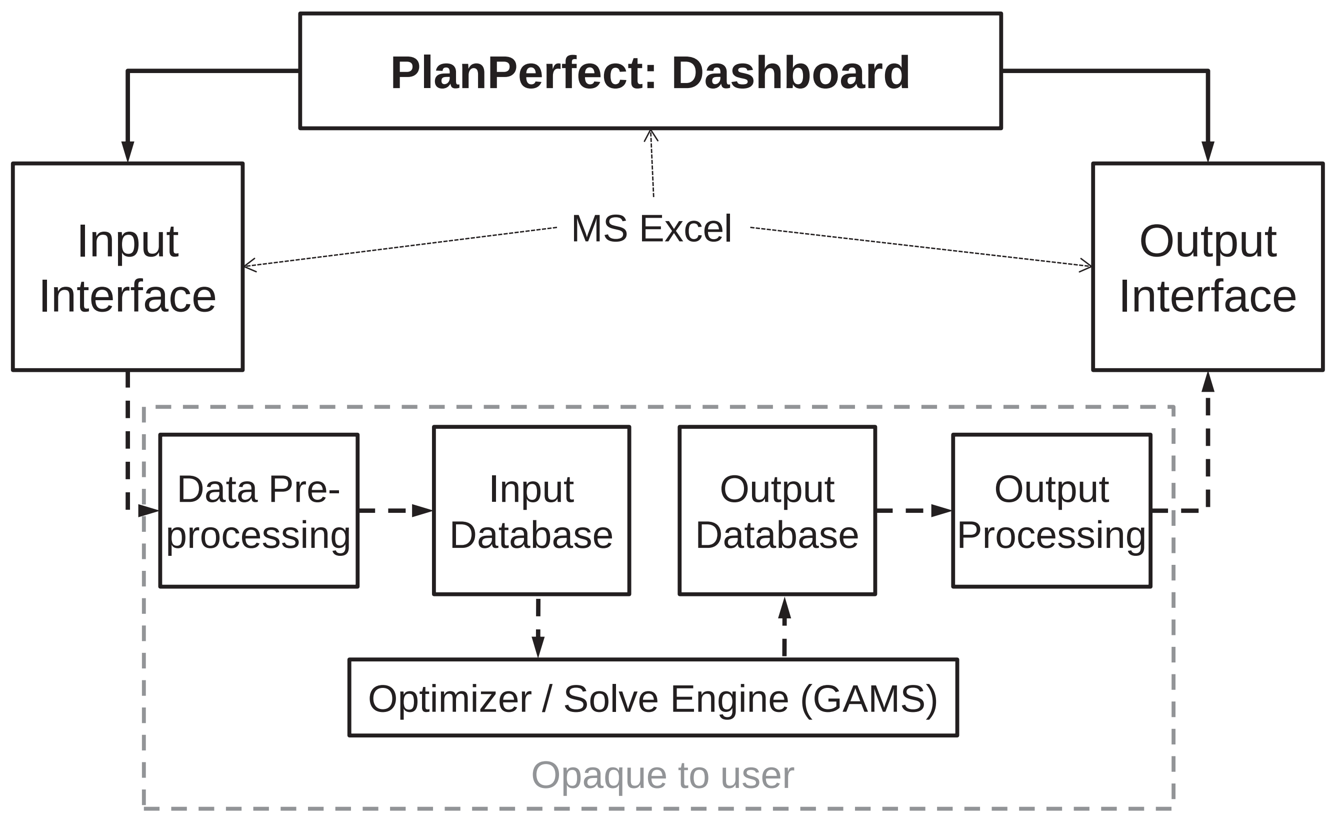

Figure 8.3 Schematic of components in PlanPerfect

1. Time horizon and intervals: Time horizon refers to the total planning period (usually, 2 or 3 years). The horizon is then divided into smaller intervals e.g. months and weeks. So, a user specifies the planning horizon in terms of months and weeks. In addition, the authors specify time related process parameters such as time required for changeovers between stages and products, minimum campaign lengths, processing time, and cycle time. Minimum campaign length gives the minimum number of weeks or minimum number of batches for which a stage should be run continuously. Such minimum numbers of weeks or batches are usually decided by the plant managers based on experience.
2. Plant configuration: Here, a user is required to define all the process equipment (production trains) and utility equipment (if applicable). Also, existing connections of production trains to utility equipment are defined.
3. Recipe: All the raw materials, intermediates, final products, and utilities should be defined at appropriate places in the tool from the drop down menu. Now, all the unit operations or tasks should be defined along with the information about the materials (utilities) consumed and produced by each task. Also, the suitability of each task over the available production lines has to be defined.
4. Process details: After defining the plant configuration and recipe information, user is required to provide the process details such as yield, batch size, and the available storage limits.
5. Scenario planning parameters: Now, for a given planning scenario user must specify the parameters such as product demand forecasts, due-dates, initial available inventories (closing stock), and operational policies (safety stock, and modes of operations as upper limits on the number of batches). Furthermore, to include the considerations for maintenance and NCEs, the user should specify

the time periods for the planned maintenance and new product testing. In addition, special requirements such as fixing some of the process operations and the resource allocation profiles such as manpower, laboratory, etc. are included at appropriate locations.

6. Costs: To evaluate the overall profit or total cost of a scenario, user is required to provide the estimates for material prices (raw materials and final products), inventory holding cost, and changeover costs. Also, to minimize the violations in meeting the demand and in maintaining safety stock, user should give suitable violation costs for each.

The first four inputs of the above are related to the plant configuration and do not change very often. Thus, such inputs are only one time inputs. The rest (scenario planning parameters and costs) are dependent on the time and keep changing. For each scenario, the user may change any of the aforementioned inputs to study its effect on the overall profit or the total cost.

Now, once all the inputs have been specified, the user has to run the optimization model by clicking the solve button on the dashboard (Figure 8.4) of the tool. A small window appears and shows the progress of the optimization (Figure 8.5). Once the solver finds a good solution either with a specified accuracy or within a specified time limit, the optimizer will stop and the small window disappears automatically. The results are then sent back to the MS-Excel workbook. In the default settings, following outputs are present in the tool, which can be modified according to the needs of the user.

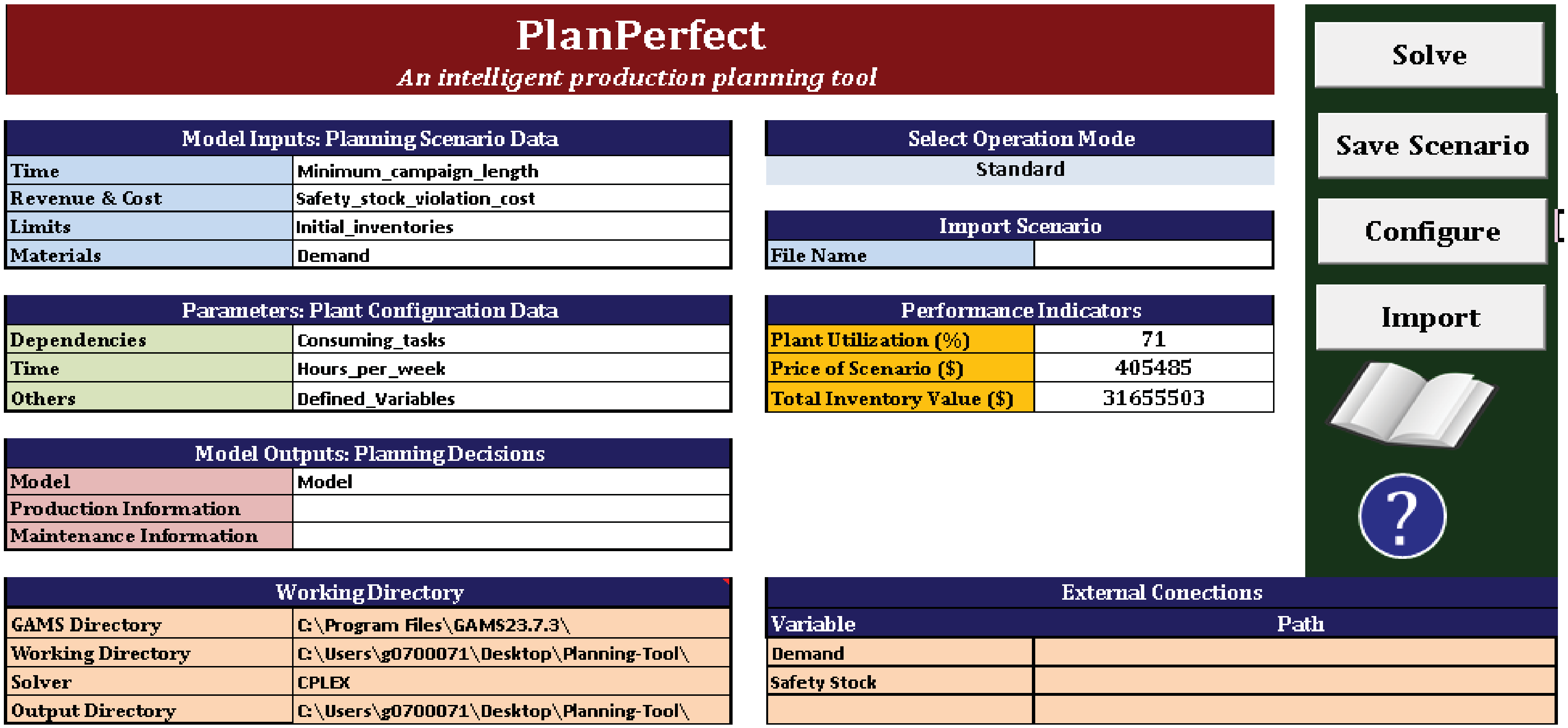

Figure 8.4 Dashboard for the GUI of PlanPerfect

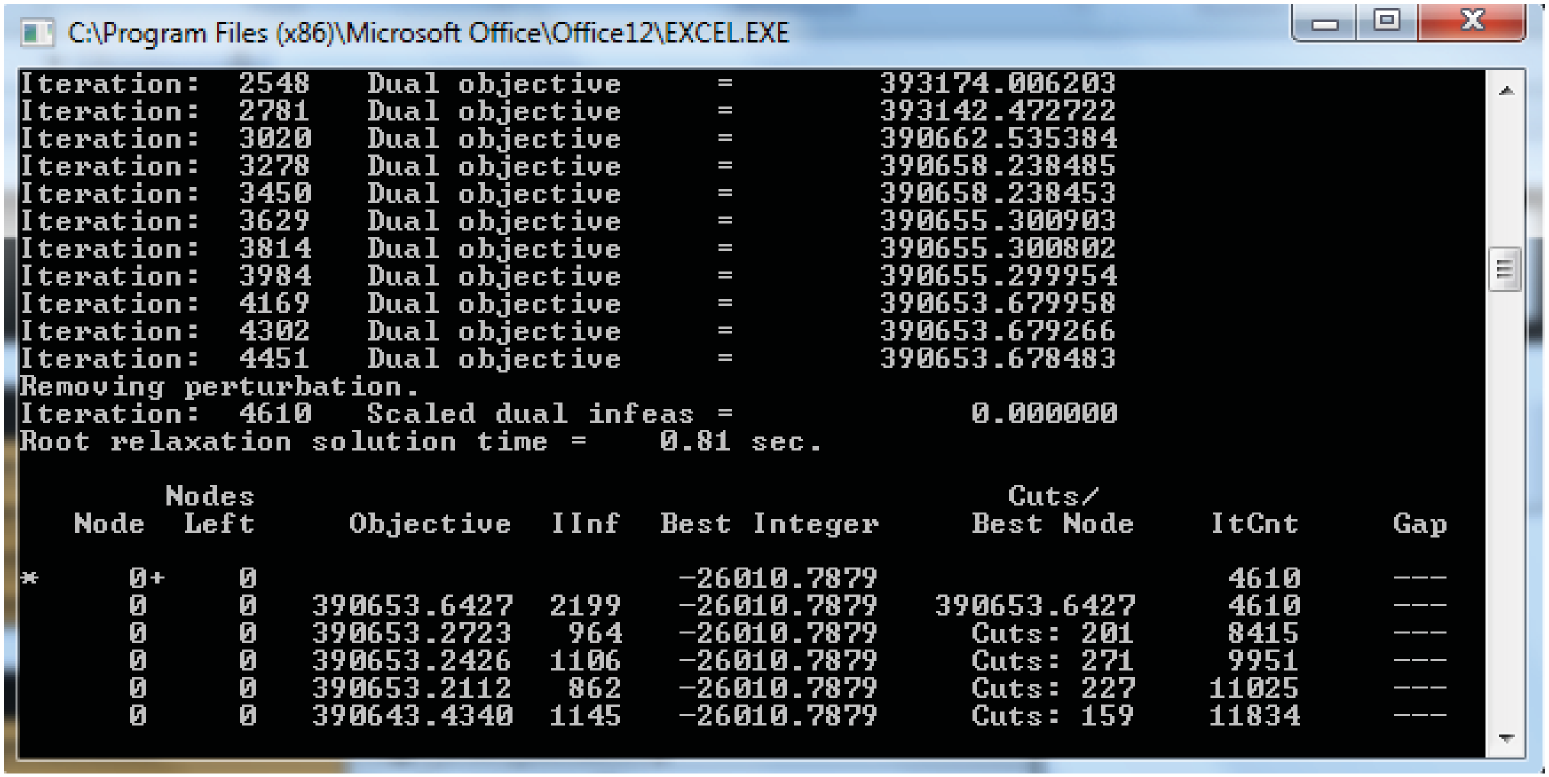

Figure 8.5 A snapshot of the intermediate pop-up window showing the progress of the optimization in the planning tool

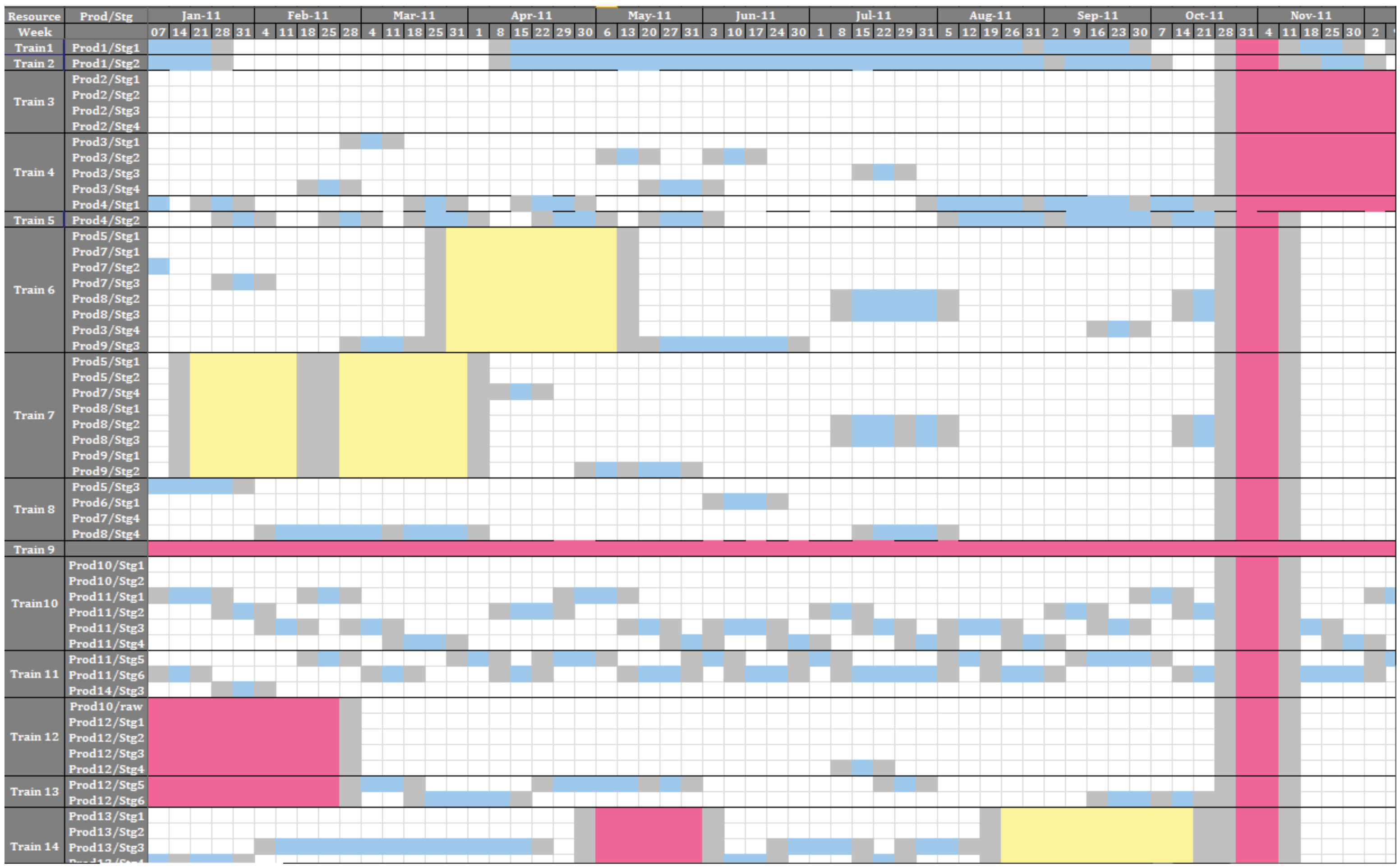

Figure 8.6 A snapshot of partial Gantt chart obtained from the tool and displaying the production plan for scenario 1.

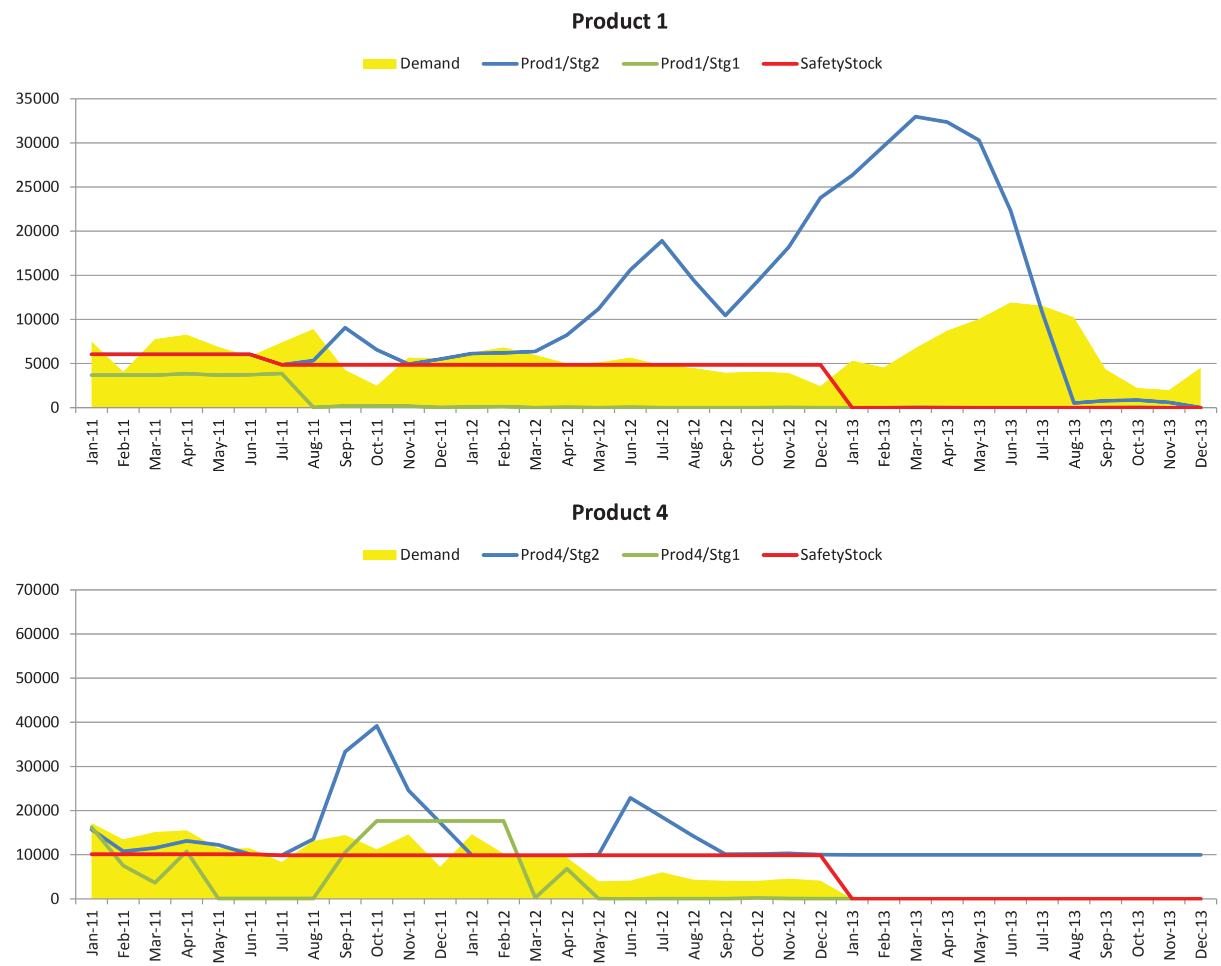

Figure 8.7 Inventory profiles for product 1 and 4 as displayed in the tool (with default settings) for scenario 1.

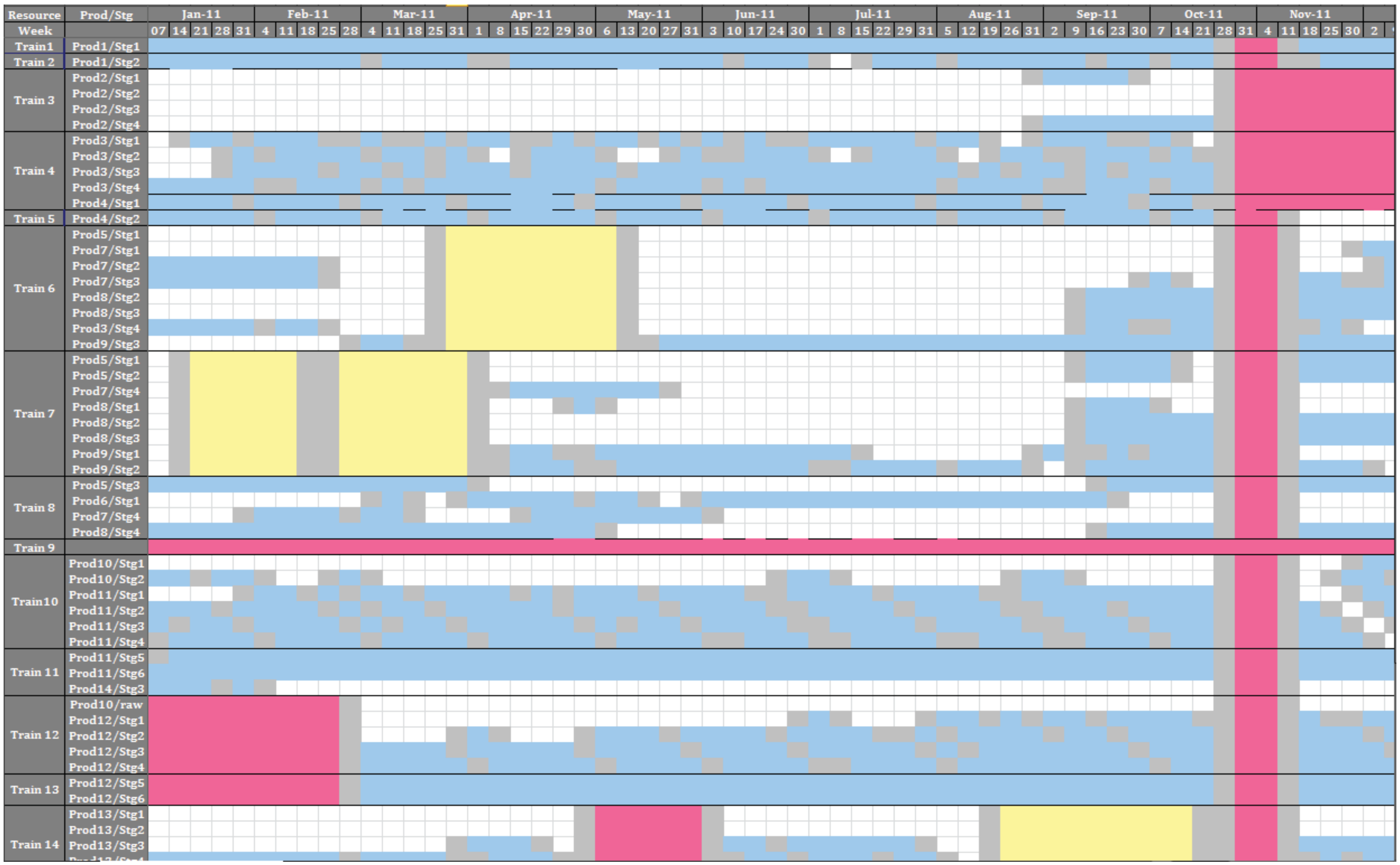

Figure 8.8 A snapshot of partial Gantt chart obtained from the tool and displaying the production plan for scenario 2.

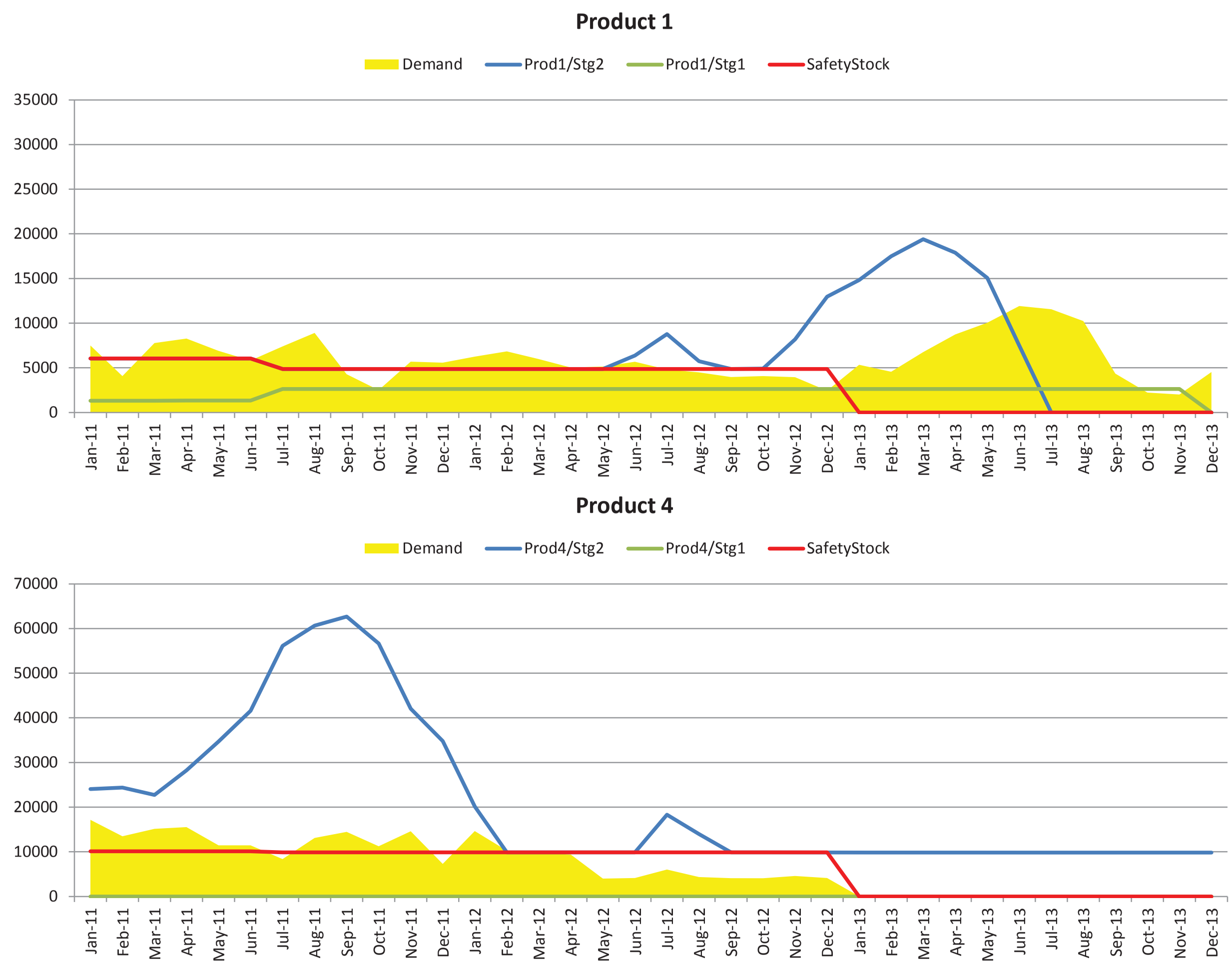

Figure 8.9 Inventory profiles for product 1 and 4 as displayed in the tool (with default settings) for scenario 2.

1. Plan: The resulting plan consisting of campaign sequences, lengths, and number of batches, changeovers, and maintenance & NCE blockages is drawn as a Gantt chart (Figure 8.6) for the specified horizon. In the Figure 8.6, the vertical axis refers to production operations on different production lines, and the horizontal axis refers to time in weeks and months. Also, the blue colour refers to the allocation of a production campaign, the red colour refers to the allocation of maintenance work, and the yellow colour refers to the allocation of NCE testing to a production line. The plan also contains a Gantt chart for all the resources allocated/used through the entire planning horizon.
2. Inventory profiles: Inventory profiles (Figure 8.7) for raw materials, intermediates, and final products are segregated into groups of product families and are plotted along with the safety stock limits for each interval. Also, the actual values of the inventories at the end of each interval are shown in a different worksheet. In Figure 8.7, the vertical axis refers to the amount of inventory and the horizontal axis refers to the user-defined time intervals such as months.

Now, to export the results, user needs to click on the export output button (Figure 8.4). This copies all input and output worksheets to a new workbook and saves it with a given name and at any location. This completes planning for one scenario with our tool.

As MS-Excel is highly popular among the potential users, it is easier for them to learn and use the tool quickly without any special training. Another useful feature of this tool is that it can be linked to other ‘workbooks’ of different departments such as sales, suppliers, maintenance, R&D, warehouse, and laboratory, from (to) where the inputs (outputs) can be imported (exported), thus enhancing collaboration, which is a vital element in production planning.

8.4 Features

One of the important characteristics of PlanPerfect is that it is specially designed and tailored for application to the pharmaceutical plants. This copyrighted tool is developed in association with a Singapore-based plant of a multi-national pharmaceutical company. It is constructed to have a competitive advantage on intelligence, speed, versatility, cost, and ease-of-use. Given the plant configuration data and planning scenario data, PlanPerfect aids decisions such as campaign scheduling, inventory management, resource allocation, material procurement, product or intermediate outsourcing, cleaning and set-ups, waste generation, and plant utilization. The tool features constraints to schedule maintenance planning, new product testing, different modes of operation, and resource allocations that are specific to the pharmaceutical companies. For production planning, PlanPerfect ensures inventory holding limits and safety stock limits. Also, it allows user to add a new or modify an old constraint easily from the interactive GUI. Although gross profit is the most comprehensive planning objective, PlanPerfect allows the planner to specify additional planning objectives such as minimum waste, maximum plant utilization, minimum production time, minimum cost, maximum order fulfilment, etc.

PlanPerfect provides user a complete freedom in changing various plant parameters and configurations. This helps user to generate a number of different planning scenarios based on different values of parameters and various plant configurations. Thus, the tool facilitates a rigorous scenario planning and then allows the evaluation of such different plans based on the performance indicators. The performance indicators can either be the value of the chosen objective such as cost, profit, inventory, etc. or they may, in addition, also include violation of safety stock limits, demand satisfaction, plant utilization, etc. Finally, depending on the type of scenario PlanPerfect gives a good solution with a pre-specified accuracy with few minutes.

Another key feature of PlanPerfect is its smart GUI (Figure 8.4). The ergonomics of the GUI is specifically customized keeping in view the requirements of planners and other stakeholders in pharmaceutical plants. The tool encompasses an interactive help (in addition to user's manual) system to navigate and assist the user. PlanPerfect offers a great flexibility to the user in terms of changing the plant configuration such that the user can add or remove any item to the tool without much effort. The tool can configure itself immediately to address the new changes. PlanPerfect has in-built feature to export and import planning scenarios at any time. Also, it can be connected to any available source such as Master Datasheet from SAP's ERP, proprietary data logs, etc. to import planning scenario data directly to the tool. Furthermore, the tool embodies a rigorous and interactive error handling system to detect, diagnose, and mitigate possible human errors.

8.5 Demonstration

Now, to demonstrate the performance of our tool we solve an industrial-scale case study. This example is motivated and modified from a real-life problem of a pharmaceutical plant. Here, we consider an active pharmaceutical ingredient manufacturing (API) plant F, involving 10 products with up to 6 stages each and handling around 48 materials. F has 42 production trains, where each train has multiple units such as mixer, reactor, separator, and drier. The plant is operated in campaigns of multiple batches. The batch size, its processing time, cycle time, and the turn-around (changeover) time are known a priori. The campaigns use a number of resources such as solvents, cooling water, steam, and operators. The amount of such utilities and number of operators required for each batch are given. Apart from materials and manpower, 4 of all the product campaigns require shared equipment such as clean rooms or dispensaries. Some products share production trains. The aforementioned information constitutes plant configuration data, which is appropriately logged in to the input database of the tool. We consider a planning horizon of 3 years. Apart from the regular products, F tests six new chemical entities (NCEs) and has a given maintenance plan during the planning horizon. Data related to the planning scenario such as product demands, safety stock limits, and minimum campaign lengths are also known and entered in the respective data sheets. Our tool allows outsourcing of products or intermediates and sourcing of raw materials. The initial stock and limits on the maximum inventory of all materials are known and are part of planning scenario data. Finally, for this example, two different operational modes (base mode and low throughput) are defined with varying resource allocation profiles. In the base mode, it is assumed that plant is operated with its complete workforce and sufficient availability of material resources. In the low throughput mode, the available number of operators is assumed to be fewer than the base case. This directly affects the operations for some of the products.

The tool determines a good production plan with suitable resource allocation profile and corresponding performance indices for two different scenarios. In scenario-1, the plant is completely operated in base mode and in scenario-2, it is completely operated in the low throughput mode. For the sake of this demonstration, we limit ourselves to only two scenarios. However, depending on user’s requirement more scenarios can be defined, where a plant can be selectively operated under a different mode for a given period of time. Here, we only consider total inventory value and plant utilization as performance indicators. User may choose to define more indicators such as total waste generation, total amount manufactured, etc. The results for both scenarios are tabulated in Table 8.1. For scenario-1, the objective value (price of scenario) of $405485 and total inventory value of $9471386 is achieved within 15 minutes of CPU time. The plant utilization for the optimal solution is found to be 66%. For scenario-2, the objective value (price of scenario) of $405485 and total inventory value of $9471386 is achieved within 15 minutes of CPU time. The plant utilization for the optimal solution is found to be 66%. The optimal plans for each scenario are represented as a Gantt chart on a calendar and inventory profiles are generated for each product. Figures 8.6 and 8.8 show the production plans for scenarios 1 and 2. Figures 8.7 and 8.9 show inventory profiles for some products.

Table 8.1 Performance indicators for both scenarios of our example.

	Scenario 1	Scenario 2
Total Inventory value ($)	31655503	31668367
Plant utilization (%)	71	82

8.6 Summary

Resource allocation and production planning are critical for efficient operations of pharmaceutical plants. In this chapter, we presented an intelligent framework for an effective and efficient production planning tool. We have captured some of the real-life challenges and constraints of the industrial planners. Also, we have tried to identify the special needs for production planning in pharmaceutical plants and highlighted the importance of resource allocation in production planning. Then, based on such a framework, we developed and presented a smart and user-friendly decision-support tool (PlanPerfect) for production planning and resource allocation. PlanPerfect is specially designed for application in pharmaceutical plants. Finally, we have briefly described the architecture and features of PlanPerfect and then successfully demonstrated its performance using an industrial-scale example.

9 CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

This thesis primarily addressed three aspects of planning and scheduling in pharmaceutical industry. These are batch and campaign scheduling in general, production planning and resource allocation in a pharmaceutical plant, and enterprise-wide operational planning in a global pharmaceutical enterprise. We developed mixed integer linear programming (MILP) based models to address planning and scheduling problems faced by the batch plants in general and pharmaceutical plants in particular. The major contributions from this research are stated as follows.

First, we presented a logical and, in our opinion, a more appropriate basis for the classification of the existing scheduling models based on the time-grids. Our classification of the existing approaches based on the number of time grids employed to model time as multi-, single-, and no-grid approaches presents a better platform for the comparison and analysis of scheduling models. Our analysis and findings from the study of some multi-grid formulations highlighted that such models – in some cases – fall short in accounting for appropriate resource balances and so, may lead to incorrect or infeasible solutions. This motivated the need for more work in the area of short-term scheduling for batch plants, especially with multi-grid approach. In this regard, we presented a detailed analysis of the basic modeling differences and challenges associated with multi-grid formulations such as handling resource balances, aligning tasks and material transfers, and the complexity of formulation due to the unavoidable big-M type constraints. We presented a fool-proof and novel framework based on unit-slots for managing shared resources such as materials and for allowing flexibility such as non-simultaneous transfers of materials into and out of a batch. We presented rational and logical arguments and constraints for the accuracy of resource balance using unit-slots. A key feature of unit slots approach is that it does not require additional binary variables for synchronizing resource balance. This is similar to the unit-specific event-based approach and thus, our approach shares similar computation benefits. In addition, for some examples, our approach requires fewer slots/events than both single- and multi-grid models that exist in the literature. This enables significant reductions in solution times and model size, and yields tighter RMIP values. We further shed light on the fact that for the worst case problems multi-grid formulations become identical to the single grid formulations. In such scenarios, the performance of multi-grid models is worse than single-grid models. This demonstrates the limitation of the current multi-grid models. Also, we highlighted the importance of constraint sequencing in GAMS implementation for evaluating MILP-based scheduling models.

Then, we generalized the concept of our multi-grid approach based on unit-slots to make it more comprehensive and closer to the real life problems by incorporating features like flexible timings for material transfer into and out of a batch, sequence-dependent cleaning times, maintenance, and utility consumption monitoring. We laid a clear understanding, from a modelling perspective, on the advantages and limitations of both single and multi-grid formulations. We highlighted that allowing non-zero transfer times and non-simultaneous material transfers using a multi-grid approach offer higher flexibility to model and handle timing constraints. On the other hand, single-grid approach tends to become highly complex in this scenario. A key finding is the importance of integrating resource constraints along with production scheduling and studying their impact on the process performance.

Next, we presented a novel framework for the production planning and the resource allocation in a pharmaceutical plant. We capture some industrially important aspects of planning activity including sequence-dependent changeovers, maintenance, NPIs, resource allocations, safety stocks, and delayed material deliveries. The major contribution here is the inception of a key idea that the resource allocation directly impacts plant’s productivity and thus, its performance. We highlighted the importance and studied the impact of integrating resource allocation decisions with the production planning.

We then extended the concept of integrating resource allocation decisions with production planning from the scale of one plant to a global enterprise. We presented a single framework, which considered operational and production decisions for the entire supply chain in a seamless fashion with a granularity of individual product campaigns. The importance of our approach lies in the fact that it accounts for practical features including effects of international tax differentials, inventory holding costs, material shelf-lives, waste treatment / disposal, and duty drawbacks.

Finally, our tool presented an intelligent and analytical approach for industrial planning activity. It embodies and illustrates a number of industrially important and user-specified features. In the form of a tool we described a smart framework to rationalize the dependency of plant productivity on resource availability. Furthermore, we highlighted the limitations of the existing technologies and established a set of features that are industrially important for such a tool.

9.2 Recommendations

During the development and evaluation of models and the study of different problems, some key points and gaps can be observed. Combined with those observations, other recommendations are as follows.

1. In chapters 3, 4, and 5, a few issues in handling resources while addressing the problem of batch scheduling using multi-grid approaches were identified and discussed. One of them was not allowing the production tasks to span over multiple slots or event-points. To address this, a novel approach of unit-slots was developed, where all tasks were allowed to span any number of slots. This, along with some appropriate task aligning constraints allows proper resource balance across multiple time-grids. For this task aligning, a critical assumption was that we can liberally insert slots of length equal to zero at any point on any resource. However, in principle, it is possible to align tasks to accurately account for resources even with the constraint that a task is allocated to only one slot. For this, one needs to develop a hybrid formulation where the tasks are allowed to span only on the resource time-grids and not on the grids of processing units. This can, potentially, reduce the number of slots/ events for some examples and thus, will be more efficient in batch scheduling.
2. There is a significant research interest towards the continuous production of pharmaceutical products. In such a scenario, it is highly likely that a plant is so designed that it partly operates in batch mode and partly in continuous or semi-continuous mode. There are a few models in the literature that present scheduling models for such hybrid plants. In this regard, a model based on the unit-slot framework may be helpful and also prove useful.

3. In chapter 6, the prime focus has been on developing an integrated model for production planning. Here, we only studied the effect of resource allocation on production planning. However, there are a few more factors that affect the planning and are critical. One of such factors is the waste generation. Our current model accommodates wastes as a product and can be constrained by the capacity of the downstream. However, in some real operations, waste treatment incurs cost, which is dependent on the type and the amount of waste treated. It may so happen that a production plan at certain times results in a high load on the waste treatment facility. To respect the emission and waste limits, operations may be forced to adjust in undesirable manner that could result in extra waste, incineration, fresh water consumption, etc. The unavailability of storage space and overload on the capacity of waste treatment facility may also occur. This is mainly because the usual industry practice is to treat production planning separate from waste treatment. Thus, an integrated study of waste treatment scheduling model with the production planning model will be useful.
4. In chapter 7, we addressed an operational planning problem for a pharmaceutical enterprise. We considered the production supply chain from the procurement of raw materials to the distribution of products to end users. One of the objectives in this study was also to minimize the inventory across the supply chain. This is to release a lot of working capital, which is locked up as inventory otherwise. In this regard, one of the important areas to reduce inventory is during the development phase of a drug. Typically, companies tend to produce the potential new products in excess. This is mainly to meet the consumer demand immediately after the completion of the clinical trials and approval of FDA. However, it is known that the number of final approvals

from the entire set of potential drugs is very low. This causes a tremendous wastage of the inventory of rejected drugs. Thus, a study of inventory minimization during the development phase of a drug and then, its integration with the global planning model is of high importance for pharmaceutical companies.

5. In chapter 8, we discussed a framework for an integrated production planning and resource allocation tool. We presented and discussed the features of one such tool. We understand that our tool can be further enhanced to improve its robustness and impart intelligence for safeguarding against uncertain operational scenarios. One of the important features of our tool, as also discussed in the chapter, is the specially designed GUI. We believe that the ergonomics of the GUI can be further enhanced to make it more user-friendly. Also, the representation of results can be made more interactive in nature. Finally, more robust approaches to deal with problem infeasibilities are needed to be explored and integrated with the tool.

REFERENCES

1. UNDESA, D.o.E.a.S.A., *The Millennium Development Goals Report 2010*2010: United Nations Publications.
2. Coopers, P.W., *Pharma 2020: The Vision–which path will you take*, 2007, Price Waterhouse Coopers, London.
3. Division, U.N.P., *World population prospects: the 2000 revision. Comprehesive tables*. Vol. 1. 2001: United Nations.
4. *Pharmaceutical Industry Outlook*. Zacks Equity Research, 2011. **March**.
5. Hofman, D., K. O'Marah, and C. Elvy, *The Gartner Supply Chain Top 25 for 2011-Demand-driven principles continue to define the leaders in this year's rankings.* World Trade, 2011. **24**(9): p. 22.
6. Gagnon, M.-A. and J. Lexchin, *The Cost of Pushing Pills: A New Estimate of Pharmaceutical Promotion Expenditures in the United States.* PLoS Med, 2008. **5**(1): p. e1.
7. Nickerson, J., *Pharmaceutical industry wastes $50 billion a year due to inefficient manufacturing.* http://news-info.wustl.edu/tips/page/normal/7912.html, 2006.
8. Suresh, P. and P.K. Basu, *Improving pharmaceutical product development and manufacturing: Impact on cost of drug development and cost of goods sold of pharmaceuticals.* Journal of Pharmaceutical Innovation, 2008. **3**(3): p. 175-187.
9. Salveson, M.E., *On a quantitative method in production planning and scheduling.* Econometrica: Journal of the Econometric Society, 1952: p. 554-590.
10. Eilon, S., *Scheduling for batch production.* Production Engineers Journal, Institution of, 1957. **36**(9): p. 549-570.
11. Reklaitis, G., *Overview of Scheduling and Planning of Batch Process Operations.* NATO Advanced Study Institute, 1992. **Batch Process System Engineering**(Antalya, Turkey).
12. Floudas, C.A. and X. Lin, *Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review.* Computers and Chemical Engineering, 2004. **28**(11): p. 2109-29.
13. McDonald, C.M. and I.A. Karimi, *Planning and scheduling of parallel semicontinuous processes. 1. Production planning.* Industrial & Engineering Chemistry Research, 1997. **36**(7): p. 2691-2700.
14. Méndez, C.A., et al., *State-of-the-art review of optimization methods for short-term scheduling of batch processes.* Computers and Chemical Engineering, 2006. **30**(6-7): p. 913-946.
15. Maravelias, C.T. and C. Sung, *Integration of production planning and scheduling: Overview, challenges and opportunities.* Computers & Chemical Engineering, 2009. **33**(12): p. 1919-1930.
16. He, Y. and C.W. Hui, *Genetic algorithm for large-size multi-stage batch plant scheduling.* Chemical Engineering Science, 2007. **62**(5): p. 1504-1523.

17. He, Y. and C.W. Hui, *Automatic rule combination approach for single-stage process scheduling problems.* AIChE Journal, 2007. **53**(8): p. 2026-2047.
18. He, Y. and C.W. Hui, *Genetic algorithm based on heuristic rules for high-constrained large-size single-stage multi-product scheduling with parallel units.* Chemical Engineering and Processing: Process Intensification, 2007. **46**(11): p. 1175-1191.
19. He, Y. and C.W. Hui, *A rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units.* Computers & Chemical Engineering, 2008. **32**(12): p. 3067-3083.
20. He, Y. and C.W. Hui, *A binary coding genetic algorithm for multi-purpose process scheduling: A case study.* Chemical Engineering Science, 2010. **65**(16): p. 4816-4828.
21. Jou, C., *A genetic algorithm with sub-indexed partitioning genes and its application to production scheduling of parallel machines.* Computers & Industrial Engineering, 2005. **48**(1): p. 39-54.
22. Azzaro-Pantel, C., et al., *A two-stage methodology for short-term batch plant scheduling:: discrete-event simulation and genetic algorithm.* Computers & Chemical Engineering, 1998. **22**(10): p. 1461-1481.
23. Ku, H.M. and I.A. Karimi, *An evaluation of simulated annealing for batch process scheduling.* Industrial & Engineering Chemistry Research, 1991. **30**(1): p. 163-169.
24. Raaymakers, W. and J. Hoogeveen, *Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing.* European Journal of Operational Research, 2000. **126**(1): p. 131-151.
25. Patel, A., R. Mah, and I. Karimi, *Preliminary design of multiproduct noncontinuous plants using simulated annealing.* Computers & Chemical Engineering, 1991. **15**(7): p. 451-469.
26. Shelokar, P., V. Jayaraman, and B. Kulkarni, *An ant colony classifier system: application to some process engineering problems.* Computers & Chemical Engineering, 2004. **28**(9): p. 1577-1584.
27. Jayaraman, V., et al., *Ant colony framework for optimal design and scheduling of batch plants.* Computers and Chemical Engineering, 2000. **24**(8): p. 1901-1912.
28. Heinonen, J. and F. Pettersson, *Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem.* Applied Mathematics and Computation, 2007. **187**(2): p. 989-998.
29. Liu, B., et al., *An effective hybrid particle swarm optimization for batch scheduling of polypropylene processes.* Computers & Chemical Engineering, 2010. **34**(4): p. 518-528.
30. Bowman, E.H., *The schedule-sequencing problem.* Operations Research, 1959: p. 621-624.
31. Bassett, M.H., J.F. Pekny, and G.V. Reklaitis, *Decomposition techniques for the solution of large-scale scheduling problems.* AIChE Journal, 1996. **42**(12): p. 3373-3387.
32. Manne, A.S., *On the job-shop scheduling problem.* Operations Research, 1960: p. 219-223.
33. Pritsker, A.A.B., L.J. Watters, and P.M. Wolfe, *Multiproject scheduling with limited resources: A zero-one programming approach.* Management science, 1969: p. 93-108.

34. Kondili, E., C.C. Pantelides, and R.W.H. Sargent, *A general algorithm for short-term scheduling of batch operations--I. MILP formulation.* Computers & chemical engineering, 1993. **17**(2): p. 211-227.
35. Shah, N., C. Pantelides, and R. Sargent, *A general algorithm for short-term scheduling of batch operations--II. Computational issues.* Computers & Chemical Engineering, 1993. **17**(2): p. 229-244.
36. Pantelides, C.C. *Unified frameworks for optimal process planning and scheduling*. in *Proceedings of the second international conference on foundations of computer-aided process operations*. 1993. Colorado: Crested Butte: Cache Publications New York.
37. Dedopoulos, I.T. and N. Shah, *Optimal short-term scheduling of maintenance and production for multipurpose plants.* Industrial & Engineering Chemistry Research, 1995. **34**(1): p. 192-201.
38. Pekny, J.F. and M.G. Zentner. *Learning to solve process scheduling problems: The role of rigorous knowledge acquisition frameworks*. in *Proceedings of the second international conference on foundations of computer-aided process operations*. 1993. Colorado: Crested Butte.
39. Zentner, M., et al., *Practical considerations in using model-based optimization for the scheduling and planning of batch/semicontinuous processes.* Journal of Process Control, 1994. **4**(4): p. 259-280.
40. Elkamel, A., et al., *A decomposition heuristic for scheduling the general batch chemical plant.* Engineering Optimization+ A35, 1997. **28**(4): p. 299-330.
41. Pitty, S.S. and I.A. Karimi, *Novel MILP models for scheduling permutation flowshops.* Chemical Product and Process Modeling, 2008. **3**(1): p. 22.
42. Méndez, C.A. and J. Cerdá, *Optimal scheduling of a resource-constrained multiproduct batch plant supplying intermediates to nearby end-product facilities.* Computers & chemical engineering, 2000. **24**(2-7): p. 369-376.
43. Hui, C.-W., A. Gupta, and H.A.J. van der Meulen, *A novel MILP formulation for short-term scheduling of multi-stage multi-product batch plants with sequence-dependent constraints.* Computers & chemical engineering, 2000. **24**(12): p. 2705-2717.
44. Gupta, S. and I.A. Karimi, *An improved MILP formulation for scheduling multiproduct, multistage batch plants.* Industrial and Engineering Chemistry Research, 2003. **42**(11): p. 2365-2380.
45. Gupta, S. and I.A. Karimi, *Scheduling a two-stage multiproduct process with limited product shelf life in intermediate storage.* Industrial and Engineering Chemistry Research, 2003. **42**(3): p. 490-508.
46. Liu, Y. and I.A. Karimi, *Scheduling multistage, multiproduct batch plants with nonidentical parallel units and unlimited intermediate storage.* Chemical Engineering Science, 2007. **62**(6): p. 1549-1566.
47. Mendez, C.A., G.P. Henning, and J. Cerda, *An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities.* Computers and Chemical Engineering, 2001. **25**(4-6): p. 701-711.
48. Mendez, C.A. and J. Cerda, *An MILP continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies.* Optimization and Engineering, 2003. **4**: p. 7-22.
49. Mendez, C.A. and J. Cerda, *Short-term scheduling of multistage batch processes subject to limited finite resources.* Computer-aided Chemical Engineering, 2004. **15 B**: p. 984-989.

50. Ferrer-Nadal, S., et al., *Material transfer operations in batch scheduling. A critical modeling issue.* Industrial & Engineering Chemistry Research, 2008. **47**(20): p. 7721-7732.
51. Wagner, H., *An integer linear-programming model for machine scheduling.* Naval Research Logistics Quarterly, 1959. **6**(2).
52. Ku, H.M., D. Rajagopalan, and I.A. Karimi, *Scheduling in batch processes.* Chemical Engineering Progress, 1987. **83**(8): p. 35-45.
53. Ku, H.M. and I.A. Karimi, *Scheduling in serial multiproduct batch processes with finite interstage storage: mixed integer linear program formulation.* Industrial and Engineering Chemistry Research, 1988. **27**(10): p. 1840-1848.
54. Ku, H.M. and I.A. Karimi, *Scheduling in serial multiproduct batch processes with due-date penalties.* Industrial & Engineering Chemistry Research, 1990. **29**(4): p. 580-590.
55. Birewar, D.B. and I.E. Grossmann, *Simultaneous production planning and scheduling in multiproduct batch plants.* Industrial & Engineering Chemistry Research, 1990. **29**(4): p. 570-580.
56. Geoffrion, A. and G. Graves, *Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach.* Operations Research, 1976: p. 595-610.
57. Sahinidis, N.V. and I.E. Grossmann, *MINLP model for cyclic multiproduct scheduling on continuous parallel lines.* Computers and Chemical Engineering, 1991. **15**(2): p. 85-103.
58. Birewar, D.B. and I.E. Grossmann, *Incorporating scheduling in the optimal design of multiproduct batch plants.* Computers & chemical engineering, 1989. **13**(1-2): p. 141-161.
59. Erdirik-Dogan, M. and I.E. Grossmann, *Slot-based formulation for the short-term scheduling of multistage, multiproduct batch plants with sequence-dependent changeovers.* Industrial and Engineering Chemistry Research, 2008. **47**(4): p. 1159-1163.
60. Karimi, I.A. and C.M. McDonald, *Planning and scheduling of parallel semicontinuous processes. 2. Short-term scheduling.* Industrial & Engineering Chemistry Research, 1997. **36**(7): p. 2701-2714.
61. Lamba, N. and I.A. Karimi, *Scheduling parallel production lines with resource constraints. 2. Decomposition algorithm.* Industrial and Engineering Chemistry Research, 2002. **41**(4): p. 790-800.
62. Lamba, N. and I.A. Karimi, *Scheduling parallel production lines with resource constraints. 1. Model formulation.* Industrial and Engineering Chemistry Research, 2002. **41**(4): p. 779-789.
63. Lim, M.-F. and I.A. Karimi, *Resource-Constrained Scheduling of Parallel Production Lines Using Asynchronous Slots.* Industrial and Engineering Chemistry Research, 2003. **42**(26): p. 6832-6842.
64. Liu, Y. and I.A. Karimi, *Novel continuous-time formulations for scheduling multi-stage batch plants with identical parallel units.* Computers and Chemical Engineering, 2007. **31**(12): p. 1671-1693.
65. Liu, Y. and I.A. Karimi, *Scheduling multistage batch plants with parallel units and no interstage storage.* Computers and Chemical Engineering, 2008. **32**(4-5): p. 671-693.
66. Pinto, J.M. and I.E. Grossmann, *Optimal cyclic scheduling of multistage continuous multiproduct plants.* Computers and Chemical Engineering, 1994. **18**(9): p. 797-816.

67. Pinto, J.M. and I.E. Grossmann, *A continuous-time mixed-integer linear-programming model for short-term scheduling of multistage batch plants.* Industrial and Engineering Chemistry Research, 1995. **34**(9): p. 3037-3051.
68. Reddy, P.C.P., I.A. Karimi, and R. Srinivasan, *A new continuous-time formulation for scheduling crude oil operations.* Chemical Engineering Science, 2004. **59**(6): p. 1325-1341.
69. Schilling, G. and C. Pantelides, *A simple continuous-time process scheduling formulation and a novel solution algorithm.* Computers and Chemical Engineering, 1996. **20**: p. 1221-1226.
70. Sundaramoorthy, A. and I.A. Karimi, *A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants.* Chemical Engineering Science, 2005. **60**(10): p. 2679-702.
71. Li, J., I. Karimi, and R. Srinivasan, *Recipe determination and scheduling of gasoline blending operations.* AIChE Journal, 2010. **56**(2): p. 441-465.
72. Zhang, X., *Algorithms for optimal scheduling using non-linear models.* Ph. D. Thesis, University of London, 1995.
73. Zhang, X. and R. Sargent, *The optimal operation of mixed production facilities—a general formulation and some approaches for the solution.* Computers and Chemical Engineering, 1996. **20**(6-7): p. 897-904.
74. Ierapetritou, M.G. and C.A. Floudas, *Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes.* Industrial and Engineering Chemistry Research, 1998. **37**(11): p. 4341-4359.
75. Castro, P.M., et al., *Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes.* Industrial and Engineering Chemistry Research, 2004. **43**(1): p. 105-118.
76. Maravelias, C.T. and I.E. Grossmann, *New general continuous-time state - Task network formulation for short-term scheduling of multipurpose batch plants.* Industrial and Engineering Chemistry Research, 2003. **42**(13): p. 3056-3074.
77. Castro, P.M., A.P. Barbosa-Póvoa, and A.Q. Novais, *Simultaneous design and scheduling of multipurpose plants using resource task network based continuous-time formulations.* Industrial & Engineering Chemistry Research, 2005. **44**(2): p. 343-357.
78. Shaik, M.A. and C.A. Floudas, *Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework.* Computers and Chemical Engineering, 2008. **32**(1-2): p. 260-74.
79. Castro, P.M. and I.E. Grossmann, *New continuous-time MILP model for the short-term scheduling of multistage batch plants.* Industrial and Engineering Chemistry Research, 2005. **44**(24): p. 9175-9190.
80. Janak, S.L. and C.A. Floudas, *Improving unit-specific event based continuous-time approaches for batch processes: Integrality gap and task splitting.* Computers and Chemical Engineering, 2008. **32**(4-5): p. 913-955.
81. Janak, S.L., X. Lin, and C.A. Floudas, *Enhanced Continuous-Time Unit-Specific Event-Based Formulation for Short-Term Scheduling of Multipurpose Batch Processes: Resource Constraints and Mixed Storage Policies.* Industrial and Engineering Chemistry Research, 2004. **43**(10): p. 2516-2533.
82. Shaik, M. and C. Floudas, *Novel Unified Modeling Approach for Short-Term Scheduling.* Industrial and Engineering Chemistry Research, 2009. **48**(6): p. 2947-2964.

83. Castro, P. and A. Novais, *Short-Term Scheduling of Multistage Batch Plants with Unlimited Intermediate Storage.* Industrial and Engineering Chemistry Research, 2008. **47**(16): p. 6126-6139.
84. Méndez, C.A., et al., *State-of-the-art review of optimization methods for short-term scheduling of batch processes.* Computers and Chemical Engineering, 2006. **30**(6-7): p. 913-946.
85. Shaik, M.A., S.L. Janak, and C.A. Floudas, *Continuous-Time Models for Short-Term Scheduling of Multipurpose Batch Plants: A Comparative Study*. Ind. Eng. Chem. Res., 2006. **45**(18): p. 6190-6209.
86. Li, J., et al., *An Analysis of Some Unit-Specific Event-Based Models for the Short-Term Scheduling of Noncontinuous Processes.* Ind. Eng. Chem. Res, 2010. **49**(2): p. 633-647.
87. Mauderli, A. and D.W.T. Rippin, *Production planning and scheduling for multi-purpose batch chemical plants.* Computers and Chemical Engineering, 1979. **3**(1): p. 199-206.
88. Wellons, M. and G. Reklaitis, *Optimal schedule generation for a single-product production line--I. Problem formulation.* Computers and Chemical Engineering, 1989. **13**(1): p. 201-212.
89. Wellons, M. and G. Reklaitis, *Scheduling of multipurpose batch chemical plants. 2. Multiple-product campaign formation and production planning.* Industrial and Engineering Chemistry Research, 1991. **30**(4): p. 688-705.
90. Wellons, M. and G. Reklaitis, *Scheduling of multipurpose batch chemical plants. 1. Formation of single-product campaigns.* Industrial and Engineering Chemistry Research, 1991. **30**(4): p. 671-688.
91. Shah, N. and C. Pantelides, *Optimal long-term campaign planning and design of batch operations.* Industrial & Engineering Chemistry Research, 1991. **30**(10): p. 2308-2321.
92. Papageorgiou, L. and C. Pantelides, *A hierarchical approach for campaign planning of multipurpose batch plants.* Computers & chemical engineering, 1993. **17**: p. 27-32.
93. Voudouris, V. and I. Grossmann, *Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory considerations.* Ind. Eng. Chem. Res, 1993. **32**(9): p. 1962-1980.
94. Tsirukis, A., S. Papageorgaki, and G. Reklaitis, *Scheduling of multipurpose batch chemical plants with resource constraints.* Industrial & Engineering Chemistry Research, 1993. **32**(12): p. 3037-3050.
95. Papageorgiou, L. and C. Pantelides, *Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. Mathematical formulation.* Ind. Eng. Chem. Res, 1996. **35**(2): p. 488-509.
96. Rajaram, K. and U. Karmarkar, *Campaign planning and scheduling for multiproduct batch operations with applications to the food-processing industry*. Manufacturing & Service Operations Management, 2004. **6**(3): p. 253.
97. Cott, B. and S. Macchietto, *Minimizing the effects of batch process variability using online schedule modification.* Computers & chemical engineering, 1989. **13**(1-2): p. 105-113.
98. Hasebe, S., I. Hashimoto, and A. Ishikawa, *General reordering algorithm for scheduling of batch processes.* Journal of chemical engineering of Japan, 1991. **24**(4): p. 483-489.

99. Kanakamedala, K., G. Reklaitis, and V. Venkatasubramanian, *Reactive schedule modification in multipurpose batch chemical plants.* Industrial & Engineering Chemistry Research, 1994. **33**(1): p. 77-90.
100. Huercio, A., A. Espuna, and L. Puigjaner, *Incorporating on-line scheduling strategies in integrated batch production control.* Computers and Chemical Engineering, 1995. **19**: p. 609-614.
101. Sanmarti, E., et al., *A combined scheduling/reactive scheduling strategy to minimize the effect of process operations uncertainty in batch plants.* Computers and Chemical Engineering, 1996. **20**: p. 1263-1268.
102. Rodrigues, M., et al., *Reactive scheduling approach for multipurpose chemical batch plants.* Computers and Chemical Engineering, 1996. **20**: p. 1215-1220.
103. Honkomp, S., L. Mockus, and G. Reklaitis, *A framework for schedule evaluation with processing uncertainty.* Computers and Chemical Engineering, 1999. **23**(4-5): p. 595-609.
104. Elkamel, A. and A. Mohindra, *A rolling horizon heuristic for reactive scheduling of batch process operations.* Engineering Optimization, 1999. **31**(6): p. 763-792.
105. Vin, J. and M. Ierapetritou, *A new approach for efficient rescheduling of multiproduct batch plants.* Ind. Eng. Chem. Res, 2000. **39**(11): p. 4228-4238.
106. Roslof, J., et al. *An MILP-based reordering algorithm for complex industrial scheduling and rescheduling*. 2000. Elsevier Science.
107. Ruiz, D., et al., *On-line fault diagnosis system support for reactive scheduling in multipurpose batch chemical plants.* Computers and Chemical Engineering, 2001. **25**(4-6): p. 829-837.
108. Janak Stacy, L., et al., *Production scheduling of a large-scale industrial batch plant. II. Reactive scheduling.* 2006. **v 45**(n 25): p. 8253-8269.
109. Ferrer-Nadal, S., et al., *Optimal Reactive Scheduling of Manufacturing Plants with Flexible Batch Recipes.* Ind. Eng. Chem. Res, 2007. **46**(19): p. 6273-6283.
110. Rijnsdorp, J., *Integrated process control and automation*1991: Elsevier New York.
111. Wilkinson, S., N. Shah, and C. Pantelides, *Aggregate modelling of multipurpose plant operation.* Computers and Chemical Engineering, 1995. **19**: p. 583-588.
112. Harjunkoski, I. and I. Grossmann, *Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods.* Computers and Chemical Engineering, 2002. **26**(11): p. 1533-1552.
113. Jain, V. and I. Grossmann, *Algorithms for hybrid MILP/CP models for a class of optimization problems.* INFORMS Journal on computing, 2001. **13**(4): p. 258-276.
114. Lin, X., E.D. Chajakis, and C.A. Floudas, *Scheduling of tanker lightering via a novel continuous-time optimization framework.* Industrial and Engineering Chemistry Research, 2003. **42**(20): p. 4441-4451.
115. Maravelias, C., *A decomposition framework for the scheduling of single-and multi-stage processes.* Computers and Chemical Engineering, 2006. **30**(3): p. 407-420.
116. Roe, B., L. Papageorgiou, and N. Shah, *A hybrid MILP/CLP algorithm for multipurpose batch process scheduling.* Computers and Chemical Engineering, 2005. **29**(6): p. 1277-1291.

117. McKay, K.N., F.R. Safayeni, and J.A. Buzacott, *A review of hierarchical production planning and its applicability for modern manufacturing.* Production Planning & Control, 1995. **6**(5): p. 384-394.
118. Amaro, A. and A. Barbosa-Póvoa, *Planning and scheduling of industrial supply chains with reverse flows: A real pharmaceutical case study.* Computers & Chemical Engineering, 2008. **32**(11): p. 2606-2625.
119. Subrahmanyam, S., J.F. Pekny, and G.V. Reklaitis, *Decomposition Approaches to Batch Plant Design and Planning.* Ind. Eng. Chem. Res., 1996. **35**(6): p. 1866-1876.
120. Grunow, M., H. Günther, and M. Lehmann, *Campaign planning for multi-stage batch processes in the chemical industry.* Or Spectrum, 2002. **24**(3): p. 281-314.
121. Dimitriadis, A.D., N. Shah, and C.C. Pantelides, *RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants.* Computers & Chemical Engineering, 1997. **21**(Supplement 1): p. S1061-S1066.
122. Sand, G. and S. Engell, *Modeling and solving real-time scheduling problems by stochastic integer programming.* Computers and Chemical Engineering, 2004. **28**(6-7): p. 1087-1103.
123. van den Heever, S.A. and I.E. Grossmann, *A strategy for the integration of production planning and reactive scheduling in the optimization of a hydrogen supply network.* Computers and Chemical Engineering, 2003. **27**(12): p. 1813-1839.
124. Verderame, P. and C. Floudas, *Integrated Operational Planning and Medium-Term Scheduling for Large-Scale Industrial Batch Plants.* Industrial & Engineering Chemistry Research, 2008. **47**(14): p. 4845-4860.
125. Papageorgiou, L. and C. Pantelides, *Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 2. A mathematical decomposition approach.* Ind. Eng. Chem. Res, 1996. **35**(2): p. 510-529.
126. Stefansson, H., N. Shah, and P. Jensson, *Multiscale planning and scheduling in the secondary pharmaceutical industry.* AIChE journal, 2006. **52**(12).
127. Benders, J.F., *Partitioning Procedures for Solving Mixed-Variables Programming Problems.* Numerische Mathematik, 1962. **4**(1): p. 238-252.
128. Geoffrion, A., *Generalized benders decomposition.* Journal of Optimization Theory and Applications, 1972. **10**(4): p. 237-260.
129. Chen, P. and J.M. Pinto, *Lagrangean-based techniques for the supply chain management of flexible process networks.* Computers & Chemical Engineering, 2008. **32**(11): p. 2505-2528.
130. Everett III, H., *Generalized Lagrange multiplier method for solving problems of optimum allocation of resources.* Operations Research, 1963: p. 399-417.
131. Fisher, M., *The Lagrangian relaxation method for solving integer programming problems.* Management science, 1981. **27**(1): p. 1-18.
132. Guignard, M. and S. Kim, *Lagrangean decomposition: a model yielding stronger Lagrangean bounds.* Mathematical programming, 1987. **39**(2): p. 215-228.
133. Wu, D. and M. Ierapetritou, *Hierarchical approach for production planning and scheduling under uncertainty.* Chemical Engineering & Processing: Process Intensification, 2007. **46**(11): p. 1129-1140.
134. Mauderli, A. and D.W.T. Rippin, *Scheduling production in multipurpose batch plants: the Batchman program.* Chem. Eng. Progress, 1980. **76**(4): p. 37.

135. Lazaro, M., A. Espuna, and L. Puigjaner, *A comprehensive approach to production planning in multipurpose batch plants.* Computers & Chemical Engineering, 1989. **13**(9): p. 1031-1047.
136. Gupta, A. and C. Maranas, *A hierarchical Lagrangean relaxation procedure for solving midterm planning problems.* Ind. Eng. Chem. Res, 1999. **38**(5): p. 1937-1947.
137. Oh, H. and I. Karimi, *Planning production on a single processor with sequence-dependent setups part 1: determination of campaigns.* Computers & Chemical Engineering, 2001. **25**(7-8): p. 1021-1030.
138. Oh, H. and I. Karimi, *Planning production on a single processor with sequence-dependent setups. Part 2: campaign sequencing and scheduling.* Computers & Chemical Engineering, 2001. **25**(7-8): p. 1031-1043.
139. Kallrath, J., *Planning and scheduling in the process industry.* Or Spectrum, 2002. **24**(3): p. 219-250.
140. Shah, N., *Pharmaceutical supply chains: key issues and strategies for optimisation.* Computers and Chemical Engineering, 2004. **28**(6-7): p. 929-941.
141. Shah, N., *Process industry supply chains: Advances and challenges.* Computers & Chemical Engineering, 2005. **29**(6): p. 1225-1236.
142. Varma, V.A., et al., *Enterprise-wide modeling & optimization-An overview of emerging research challenges and opportunities.* Computers and Chemical Engineering, 2007. **31**(5-6): p. 692-711.
143. Suryadi, H. and L.G. Papageorgiou, *Optimal maintenance planning and crew allocation for multipurpose batch plants.* International Journal of Production Research, 2004. **42**(2): p. 355-377.
144. Sundaramoorthy, A. and I. Karimi, *Planning in pharmaceutical supply chains with outsourcing and new product introductions.* Ind. Eng. Chem. Res, 2004. **43**(26): p. 8293-8306.
145. Verderame, P. and C. Floudas, *Operational planning framework for multisite production and distribution networks.* Computers & Chemical Engineering, 2009. **33**(5): p. 1036-1050.
146. Corsano, G., P. Aguirre, and J. Montagna, *Multiperiod design and planning of multiproduct batch plants with mixed-product campaigns.* AIChE Journal, 2009. **55**(9): p. 2356-2369.
147. Elmaghraby, S., *Activity networks: Project planning and control by network models*1977: John Wiley & Sons.
148. Schmidt, C.W. and I.E. Grossmann, *Optimization Models for the Scheduling of Testing Tasks in New Product Development.* Industrial & Engineering Chemistry Research, 1996. **35**(10): p. 3498-3510.
149. Jain, V. and I.E. Grossmann, *Resource-Constrained Scheduling of Tests in New Product Development.* Industrial & Engineering Chemistry Research, 1999. **38**(8): p. 3013-3026.
150. Blau, G., et al., *Risk management in the development of new products in highly regulated industries.* Computers and Chemical Engineering, 2000. **24**(2-7): p. 659-664.
151. Maravelias, C. and I. Grossmann, *Simultaneous planning for new product development and batch manufacturing facilities.* Industrial and Engineering Chemistry Research, 2001. **40**(26): p. 6147-6164.
152. Subramanian, D., J. Pekny, and G. Reklaitis, *A simulation-optimization framework for research and development pipeline management.* AIChE journal, 2001. **47**(10).

153. Subramanian, D., et al., *Simulation-optimization framework for stochastic optimization of R&D pipeline management.* AIChE journal, 2003. **49**(1).
154. Blau, G., et al., *Managing a portfolio of interdependent new product candidates in the pharmaceutical industry*. Journal of Product Innovation Management, 2004. **21**(4): p. 227-245.
155. Varma, V., et al., *A framework for addressing stochastic and combinatorial aspects of scheduling and resource allocation in pharmaceutical R&D pipelines.* Computers and Chemical Engineering, 2008. **32**(4-5): p. 1000-1015.
156. Colvin, M. and C. Maravelias, *A stochastic programming approach for clinical trial planning in new drug development.* Computers and Chemical Engineering, 2008. **32**(11): p. 2626-2642.
157. Colvin, M. and C.T. Maravelias, *Scheduling of testing tasks and resource planning in new product development using stochastic programming.* Computers & Chemical Engineering, 2009. **33**(5): p. 964-976.
158. McDonald, C. and G. Reklaitis, *Design and Operation of Supply Chains in Support of Business and Financial Strategies.* Foundations of Computer-Aided Process Design, FOCAPD, Princeton, NJ, July, 2004: p. 11-16.
159. Grossmann, I.E., *Enterprise-wide optimization: A new frontier in process systems engineering.* AIChE Journal, 2005. **51**(7): p. 1846-1857.
160. Grossmann, I.E., *Challenges in the new millennium: product discovery and design, enterprise and supply chain optimization, global life cycle assessment.* Computers & Chemical Engineering, 2004. **29**(1): p. 29-39.
161. Barbosa-Póvoa, A.P., *Sustainable Supply Chains: Key Challenges.* Computer Aided Chemical Engineering, 2009. **27**: p. 127-132.
162. Papageorgiou, L.G., *Supply chain optimisation for the process industries: Advances and opportunities.* Computers & Chemical Engineering, 2009. **33**(12): p. 1931-1938.
163. Cohen, M.A. and H.L. Lee, *Strategic Analysis of Integrated Production-Distribution Systems: Models and Methods.* Operations Research, 1988. **36**(2): p. 216-228.
164. Timpe, C.H. and J. Kallrath, *Optimal planning in large multi-site production networks.* European Journal of Operational Research, 2000. **126**(2): p. 422-435.
165. Papageorgiou, L.G., G.E. Rotstein, and N. Shah, *Strategic supply chain optimization for the pharmaceutical industries.* Industrial & Engineering Chemistry Research, 2001. **40**(1): p. 275-286.
166. Gatica, G., L. Papageorgiou, and N. Shah, *Capacity planning under uncertainty for the pharmaceutical industry*. Chemical Engineering Research and Design, 2003. **81**(6): p. 665-678.
167. Levis, A.A. and L.G. Papageorgiou, *A hierarchical solution approach for multi-site capacity planning under uncertainty in the pharmaceutical industry*. Computers & Chemical Engineering, 2004. **28**(5): p. 707-725.
168. Oh, H.C. and I. Karimi, *Regulatory factors and capacity-expansion planning in global chemical supply chains.* Industrial & Engineering Chemistry Research, 2004. **43**(13): p. 3364-3380.
169. Oh, H.C. and I.A. Karimi, *Global multiproduct production–Distribution planning with duty drawbacks.* AIChE Journal, 2006. **52**(2): p. 595-610.
170. Sundaramoorthy, A., et al., *An Integrated Model for Planning in Global Chemical Supply Chains.* Computer Aided Chemical Engineering, 2006. **21**(B): p. 2189.

171. Ryu, J. and E.N. Pistikopoulos, *Multiperiod planning of enterprise-wide supply chains using an operation policy.* Industrial & Engineering Chemistry Research, 2007. **46**(24): p. 8058-8065.
172. Naraharisetti, P., I. Karimi, and R. Srinivasan, *Supply chain redesign through optimal asset management and capital budgeting.* Computers & Chemical Engineering, 2008. **32**(12): p. 3153-3169.
173. Bok, J.-K., I.E. Grossmann, and S. Park, *Supply Chain Optimization in Continuous Flexible Process Networks.* Industrial & Engineering Chemistry Research, 2000. **39**(5): p. 1279-1290.
174. Norton, L.C. and I.E. Grossmann, *Strategic planning model for complete process flexibility.* Industrial & Engineering Chemistry Research, 1994. **33**(1): p. 69-76.
175. Jayaraman, V. and H. Pirkul, *Planning and coordination of production and distribution facilities for multiple commodities.* European Journal of Operational Research, 2001. **133**(2): p. 394-408.
176. Park, Y.B., *An integrated approach for production and distribution planning in supply chain management.* International Journal of Production Research, 2005. **43**(6): p. 1205-1224.
177. Sousa, R., N. Shah, and L.G. Papageorgiou, *Supply chain design and multilevel planning--An industrial case.* Computers & Chemical Engineering, 2008. **32**(11): p. 2643-2663.
178. Eskigun, E., et al., *Outbound supply chain network design with mode selection, lead times and capacitated vehicle distribution centers.* European Journal of Operational Research, 2005. **165**(1): p. 182-206.
179. You, F. and I.E. Grossmann, *Integrated multi-echelon supply chain design with inventories under uncertainty: MINLP models, computational strategies.* AIChE Journal, 2010. **56**(2): p. 419-440.
180. You, F., I.E. Grossmann, and J.M. Wassick, *Multisite Capacity, Production, and Distribution Planning with Reactor Modifications: MILP Model, Bilevel Decomposition Algorithm versus Lagrangean Decomposition Scheme.* Industrial & Engineering Chemistry Research, 2010. **50**(9): p. 4831-4849.
181. Sousa, R.T., et al., *Global Supply Chain Planning for Pharmaceuticals.* Chemical Engineering Research and Design, 2011.
182. Karmarkar, N.K., *Methods and apparatus for efficient resource allocation*, in *United States Patent and Trademark Office*, USPTO, Editor 1988, American Telephone and Telegraph Company, AT&T Bell Laboratories: United States.
183. Bayer, D.A., N.K. Karmarkar, and J.C. Lagarias, *Method and apparatus for optimizing system operational parameters*, in *United States Patent and Trademark Office*, USPTO, Editor 1988, American Telephone and Telegraph Company, AT&T Bell Laboratories: United States.
184. Dembo, R.S., *Scenario optimization*, in *United States Patent and Trademark Office*, USPTO, Editor 1992: United States.
185. Kurihara, K., et al., *Work scheduling method*, in *United States Patent and Trademark Office*, USPTO, Editor 1991, Hitachi, Ltd., Hitachi Microcomputer Engineering Ltd.: United States.
186. Deziel Jr, L.B. and L. Finstad, *Method and apparatus for resource constraint scheduling*, in *United States Patent and Trademark Office*, USPTO, Editor 1995, Sun Microsystems, Inc.: United States.

187. Dietrich, B.L. and R.J. Wittrock, *Optimization of manufacturing resource planning*, in *United States Patent and Trademark Office*, USPTO, Editor 1997, International Business Machines Corporation: United States.
188. Trautmann, N. and C. Schwindt, *Method of plant occupancy planning in the process industry*, in *United States Patent and Trademark Office*, USPTO, Editor 2006, SAP AG: United State.
189. Strain, C., et al., *Automated Batch Manufacturing*, in *United States Patent and Trademark Office*, USPTO, Editor 2007, Pfizer Inc: United States.
190. Kataria, A.R., J. Prang, and V. Ambekar, *Systems and methods for managing the development and manufacturing of a drug*, in *United States Patent and Trademark Office*, USPTO, Editor 2009, Conformia Software, Inc.: United States.
191. Goodall, C., N. Jhaveri, and R.A. Wielgos, *Method and apparatus for inter-pharmacy workload balancing using resource function assignments*, in *United States Patent and Trademark Office*, USPTO, Editor 2010, Walgreen Co.: United States.
192. Popp, S.M., *Manufacturing execution system for validation, quality and risk assessment and monitoring of pharmaceutical manufacturing processes*, in *United States Patent and Trademark Office*, USPTO, Editor 2010, SMP Logic Systems LLC: United States.
193. Couronne, P., F. Focacci, and F. Paulin, *Planning Method and System for Optimizing Stock and Production Plan in Batch Process Industry*, in *United States Patent and Trademark Office*, USPTO, Editor 2011: United States.
194. Mockus, L. and G.V. Reklaitis, *Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization.* Computers & chemical engineering, 1997. **21**(10): p. 1147-1156.
195. Shah, N., C.C. Pantelides, and R.W.H. Sargent, *A general algorithm for short-term scheduling of batch operations--II. Computational issues.* Computers & chemical engineering, 1993. **17**(2): p. 229-244.
196. Castro, P., A.P.F.D. Barbosa-Povoa, and H. Matos, *An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants.* Industrial and Engineering Chemistry Research, 2001. **40**(9): p. 2059-2068.
197. Giannelos, N.F. and M.C. Georgiadis, *A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes.* Industrial and Engineering Chemistry Research, 2002. **41**(9): p. 2178-2184.
198. Giannelos, N.F. and M.C. Georgiadis, *A novel event-driven formulation for short-term scheduling of multipurpose continuous processes.* Industrial and Engineering Chemistry Research, 2002. **41**(10): p. 2431-2439.
199. Ierapetritou, M.G. and C.A. Floudas, *Effective continuous-time formulation for short-term scheduling. 2. Continuous and semicontinuous processes.* Industrial & Engineering Chemistry Research, 1998. **37**(11): p. 4360-4374.
200. Ierapetritou, M.G., T.S. Hene, and C.A. Floudas, *Effective continuous-time formulation for short-term scheduling. 3. Multiple intermediate due dates.* Industrial and Engineering Chemistry Research, 1999. **38**(9): p. 3446-3461.
201. Jia, Z. and M. Ierapetritou, *Mixed-integer linear programming model for gasoline blending and distribution scheduling.* Industrial and Engineering Chemistry Research, 2003. **42**(4): p. 825-835.

202. Jia, Z. and M. Ierapetritou, *Efficient short-term scheduling of refinery operations based on a continuous time formulation.* Computers and Chemical Engineering, 2004. **28**(6-7): p. 1001-1019.
203. Jia, Z., M. Ierapetritou, and J.D. Kelly, *Refinery short-term scheduling using continuous time formulation: Crude-oil operations.* Industrial and Engineering Chemistry Research, 2003. **42**(13): p. 3085-3097.
204. Lin, X. and C.A. Floudas, *Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation.* Computers and Chemical Engineering, 2001. **25**(4-6): p. 665-674.
205. Shaik, M.A. and C.A. Floudas, *Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: Rigorous treatment of storage requirements.* Industrial and Engineering Chemistry Research, 2007. **46**(6): p. 1764-1779.
206. Susarla, N., J. Li, and I. Karimi, *A novel approach to scheduling multipurpose batch plants using unit slots.* AIChE Journal, 2010. **56**(7): p. 1859-1879.
207. Giménez, D.M., G.P. Henning, and C.T. Maravelias, *A novel network-based continuous-time representation for process scheduling: Part I. Main concepts and mathematical formulation.* Computers & Chemical Engineering, 2009. **33**(9): p. 1511-1528.
208. Giménez, D.M., G.P. Henning, and C.T. Maravelias, *A novel network-based continuous-time representation for process scheduling: Part II. General framework.* Computers & Chemical Engineering, 2009. **33**(10): p. 1644-1660.
209. Karimi, I.A. and A. Sundaramoorthy. *An improved formulation for scheduling multipurpose batch plants using synchronous slots*. 2004. Austin, TX, United States: American Institute of Chemical Engineers, New York, NY 10016-5991, United States.
210. Brooke, A.D., D. Kendrick, and A. Meeraus, eds. *GAMS: A User's Guide*. 1988, The Scientific Press: South San Francisco, CA.
211. Karimi, I., Z. Tan, and S. Bhushan, *An improved formulation for scheduling an automated wet-etch station.* Computers and Chemical Engineering, 2004. **29**(1): p. 217-224.
212. Giménez, D.M., C.T. Maravelias, and G.P. Henninga, *A novel network-based continuous-time framework for batch process scheduling: Modeling and computational issues*. 2010.
213. Susarla, N. and I.A. Karimi, *Integrated campaign planning and resource allocation in batch plants.* Computers & Chemical Engineering, 2011. **35**(12): p. 2990-3001.
214. Basu, P., et al., *Analysis of manufacturing costs in pharmaceutical companies.* Journal of Pharmaceutical Innovation, 2008. **3**(1): p. 30-40.