OPTIMIZATION OF JUST-IN-TIME SEQUENCING PROBLEMS AND SUPPLY CHAIN LOGISTICS

Gyan Thapa

2015

School of Education, Culture and Communication
OPTIMIZATION OF JUST-IN-TIME SEQUENCING PROBLEMS AND SUPPLY CHAIN LOGISTICS

Gyan Thapa

Akademisk avhandling

som för avläggande av filosofie doktorsexamen i matematik/tillämpad matematik
vid Akademin för utbildning, kultur och kommunikation kommer att offentligen
försvaras måndagen den 12 oktober 2015, 13.15 i Kappa, Mälardalens högskola, Västerås.

Fakultetsopponent: Professor George Fodor, Q-TAGG R&D AB

MÄLARDALEN UNIVERSITY SWEDEN

Akademin för utbildning, kultur och kommunikation
Abstract

This dissertation presents a comprehensive and comparative progress in sequencing approaches of mixed-model just-in-time (JIT) sequencing problem together with discrete apportionment problem (DAP). The goal of JIT sequencing problem (JITSP) is to keep the rate of usage of parts as constant as possible along the assembly lines, and the goal of DAP is to divide a given integer number of delegates proportionally among the states or the parties according to their population or votes. Furthermore, the supply chain logistics problem is also reported in here with some real life applications. The single-level JITSP, known as the product rate variation problem (PRVP), is pseudo-polynomially solvable. The total PRVP minimizes sum deviation and the bottleneck PRVP minimizes the maximum deviation between the actual production and the ideal production. The assignment approach solves total PRVP whereas the perfect matching works for bottleneck PRVP solving the problem in pseudo-polynomial time. The multi-level JITSP, known as the output rate variation problem (ORVP), is NP-hard in most of the cases. However, some sequencing heuristics and dynamic programming are devised for near optimal solutions. And the pegging assumption reduces the ORVP into weighted case of PRVP. In this dissertation, the total PRVP with square and absolute deviations are considered and mean-based divisor methods are devised for the equitably efficient solution. The simultaneous dealing to the PRVP and DAP establishes the interlink between the production sequencing problem and integer seat allocating problem. The new upper bottlenecks are investigated and the problems are solved comparatively. The bottleneck PRVP instances for small deviations and cyclic sequences for total PRVP are shown to be optimal. The bicriterion sequencing is discussed with Pareto optimal solutions. The production sequencing problem is simultaneously dealt with supply chain logistics to balance overall supply chain system. The cross-docking supply chain logistics problem is formulated with a proposition to be solved. The real-world applications of JITSP and supply chain are listed and some open problems are pointed out as the closing of the dissertation.
This dissertation presents a comprehensive and comparative progress in sequencing approaches of mixed-model just-in-time (JIT) sequencing problem together with discrete apportionment problem (DAP). The goal of JIT sequencing problem (JITSP) is to keep the rate of usage of parts as constant as possible along the assembly lines, and the goal of DAP is to divide a given integer number of delegates proportionally among the states or the parties according to their population or votes. Furthermore, the supply chain logistics problem is also reported in here with some real life applications.

The single-level JITSP, known as the product rate variation problem (PRVP), is pseudo-polynomially solvable. The total PRVP minimizes sum deviation and the bottleneck PRVP minimizes the maximum deviation between the actual production and the ideal production. The assignment approach solves total PRVP whereas the perfect matching works for bottleneck PRVP solving the problem in pseudo-polynomial time. The multi-level JITSP, known as the output rate variation problem (ORVP), is NP-hard in most of the cases. However, some sequencing heuristics and dynamic programming are devised for near optimal solutions. And the pegging assumption reduces the ORVP into weighted case of PRVP.

In this dissertation, the total PRVP with square and absolute deviations are considered and mean-based divisor methods are devised for the equitably efficient solution. The simultaneous dealing to the PRVP and DAP establishes the interlink between the production sequencing problem and integer seat allocating problem. The new upper bottlenecks are investigated and the problems are solved comparatively. The bottleneck PRVP instances for small deviations and cyclic sequences for total PRVP are shown to be optimal. The bicriterion sequencing are discussed with Pareto optimal solutions.
The production sequencing problem is simultaneously dealt with supply chain logistics to balance overall supply chain system. The cross-docking supply chain logistics problem is formulated with a proposition to be solved. The real-world applications of JITSP and supply chain are listed and some open problems are pointed out as the closing of the dissertation.

**Key-words:** Just-in-time, JIT sequencing, apportionment, product rate variation, output rate variation, algorithms, heuristics, supply chain, logistics, queueing, non-linear integer programming
ACKNOWLEDGEMENTS

I am very grateful to my supervisor Professor Sergei Silvestrov and co-supervisors Professor Anatoliy Malyarenko, Professor Tanka N. Dhamala and Prof. Shankar Raj Pant for their scientific supervision, regular motivation, suggestions, helpful comments and sincere response to each of my queries. Their support, encouragement and scientific guidance inspired me to carry on this research project to this end.

I would like to express my sincere gratitude to IDEAS Project funded by Erasmus Mundus which provided me the opportunity to work under the grant at the Division of Applied Mathematics, The School of Education, Culture and Communication, Mälardalen University, Västerås, Sweden. I have a very good impression while working in Mälardalen University itself, so my sincere thanks go to this university as well, and equally to Institute of Engineering, Tribhuvan University, Nepal for providing me study leave for the research.

I am greatly benefited from Ann-Catrin Davis, Kristina Konpan and Carina Karlsson from their invaluable helps in administrative and secretarial matters. Similarly I am indebted to my teachers, especially to Mani Ratna Ghimire, Tika Ram Ghimire, Khadga KC, Basu Dev Aryal, Hira Mani Aryal, Om P. Chaube, Dhirendra Lal Das, Prof. Tanka N. Dhamala, Prof. Ram Man Shrestha, Prof. Hom Nath Bhattarai, Prof. Yoga Ratna Sthapit, Prof. Gajendra Bdr. Thapa, Prof. Shankar Raj Pant and Prof. Hongnian Yu.
In this moment, I heartily remember some of my friends, in particular Prof. Subarna Shakya, Prof. Tika Ram Aryal, Rameshwar Rijal, Karl Lundengård, Christopher Engström, Kashif Shahzad, Baiplu Sissi, Bal Krishna Nyaupane and Drawendra Chaudhary for their love, fruitful suggestion and many types of assistance during my research work.

Finally, I express my heart-felt love to my wife Mrs. Rena Thapa, and our daughters Agya and Abha for bearing almost all the hardships of our family life during the time interval of my research work and foreign visits.

Likewise, I thank to all those who wish me a success.
In this moment, I heartily remember some of my friends, in particular Prof. Subarna Shakya, Prof. Tika Ram Aryal, Rameshwar Rijal, Karl Lundeng˚ ard, Christopher Engstr¨ om, Kashif Shahzad, Baiplu Sissi, Bal Krishna Nyaupane and Drawendra Chaudhary for their love, fruitful suggestion and many types of assistance during my research work.

Finally, I express my heart-felt love to my wife Mrs. Rena Thapa, and our daughters Agya and Abha for bearing almost all the hardships of our family life during the time interval of my research work and foreign visits. Likewise, I thank to all those who wish me a success.

Dedicated

to

my father

Late, Gammar Singh Thapa
LIST OF SYMBOLS

- $i, j \in \{1, 2, \ldots, n\}$: model
- $k, l \in \{1, 2, \ldots, D\}$: time unit
- $d_{ij}$: demand of model $i$ in copy $j$
- $R_i$: period of job $i$
- $\tilde{R}_i$: run time of job $i$
- $D$: total demand
- $P$: total population
- $r_i$: demand rate of model $i$
- $r_{\text{max}}$: maximum demand rate of any model
- $T$: planning horizon
- $c$: cycle time
- $x_{ik}$: cumulative production of model $i$ in period $k$
- $x_{ijk}$: decision variable for $(i, j)$ at $k$
- $f_i$: non-negative unimodal convex symmetric function
- $d(a)$: divisor function
- $l, l \in \{1, 2, \ldots, L\}$: production level
- $d_{il}$: demand for part $i$ of level $l$
- $t_{ilp}$: number of total units of part $i$
LIST OF SYMBOLS

$i, i = 1, 2, \ldots, n$  
$\text{model}$

$k, k = 1, 2, \ldots, D$  
$\text{time unit}$

$(i, j)$  
$\text{$j^{th}$ copy of model $i$}$

$d_i$  
$\text{demand of model $i$}$

$R_i$  
$\text{period of job $i$}$

$\tilde{R}_i$  
$\text{run time}$

$D$  
$\text{total demand}$

$P$  
$\text{total population}$

$r_i$  
$\text{demand rate of model $i$}$

$r_{\max}$  
$\text{maximum $r_i$}$

$T$  
$\text{planning horizon}$

$c$  
$\text{cycle time}$

$x_{ik}$  
$\text{cumulative production of model $i$}$

$x_{ijk}$  
$\text{decision variable for (i, j) at k}$

$f_i$  
$\text{non-negative unimodal convex symmetric function}$

$d(a)$  
$\text{divisor function}$

$l, l = 1, 2, \ldots, L$  
$\text{production level}$

$d_{il}$  
$\text{demand for part $i$ of level $l$}$

$t_{ilp}$  
$\text{number of total units of part $i$}$

$t$  
$\text{takt-time}$

$p$  
$\text{product}$

$h$  
$\text{house size}$
$a$ apportionment
$d_{p1}$ demand for part $i$ of level 1
$D_l$ total part demands of level $l$
$r_{il}$ demand ratio
$x_{ilk}$ cumulative quantity of part $i$ produced at level $l$
$y_{lk}$ total quantity produced at level $l$
$y_{1k}$ total quantity produced at level 1
$C_i$ completion time of job $i$
$C^i_{jk}$ assignment total cost for all copies of model $i$
$\psi^i_{jk}$ assignment cost for $j^{th}$ copy of model $i$
$p_i$ processing time of job $i$
$P_i$ population of state $i$
$s_i$ setup time of job $i$
$w_l$ weight of level $l$
$w_{il}$ weight for model $i$ of level $l$
$X$ states in a schedule
$|X|$ cardinality of $X$
$\mathcal{X}$ set of the assignment of $(i, j)$ to $k$
$\phi(X)$ minimum of the maximum absolute deviation
$\Gamma$ deviation matrix
$\gamma_{ilp}$ \( \sum_{m=1}^{l-1} (n_m + 1)^{th} \) row and $p^{th}$ column element
$\Phi(X)$ minimum of the total square deviation
$\left(\|\Omega X_k\|_2\right)^2$ sum of square deviations
$\|\Gamma X\|_1$ maximum absolute deviation
$Z_{ij}$ ideal position of $j^{th}$ copy of model $i$
$\tilde{F}$ objective of bottleneck ORVP
$\tilde{G}$ objective of total ORVP
$s^m$ concatenation of $s$
$E(i, j)$ earliest sequencing time for $(i, j)$
$L(i, j)$ latest sequencing time for $(i, j)$
\( F_{\text{max}} \) objective of bottleneck PRVP
\( B \) bottleneck (bound)
\( F_{\text{sum}} \) objective of total PRVP
\( \mathcal{G} \) convex bipartite graph
\( V_1 \) set of sequencing times
\( V_2 \) set of \( j^{th} \) copy of model \( i \)
\( \mathcal{E} \) edge set
\( \mathcal{M} \) matching
\( K \) subset \( V_1 \)
\( N(K) \) neighbourhood of \( K \)
\( \mathcal{T} \) interval in \( V_1 \)
\( S_1 \) set of feasible sequences with \( B \leq 1 \)
\( F_a^{\text{max}} \) absolute-deviation objective of bottleneck PRVP
\( F_s^{\text{max}} \) square-deviation objective of bottleneck PRVP
\( F_m \) general objective of bottleneck PRVP
\( F_a^{\text{sum}} \) absolute-deviation objective of total PRVP
\( F_s^{\text{sum}} \) square-deviation objective of total PRVP
\( I \) set of inbound trucks
\( O \) set of outbound trucks
LIST OF ABBREVIATIONS

DAP: Discrete apportionment problem
DP: Dynamic programming
EDD: Earliest due date
EGCM: Extended goal chasing method
GCM: Goal chasing method
JIT: Just-in-time
JITPS: Just-in-time production system
JITSP: Just-in-time sequencing problem
MA: Miltenburg's algorithm
MA3H1: Miltenburg's algorithm 3 with heuristic 1
MA3H2: Miltenburg's algorithm 3 with heuristic 2
MBD: Mean-based divisor
MDJIT: Maximum deviation just-in-time
MEP: Method of equal proportion
NP-hard: Non-deterministic polynomial-time hard
ORVP: Output rate variation problem
PRVP: Product rate variation problem
RTVP: Response time variability problem
SCL: Supply chain logistics
SDC: Small deviation conjecture
SDJIT: Sum deviation just-in-time
TRSP: Truck sequencing problem
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAP</td>
<td>Discrete apportionment problem</td>
</tr>
<tr>
<td>DP</td>
<td>Dynamic programming</td>
</tr>
<tr>
<td>EDD</td>
<td>Earliest due date</td>
</tr>
<tr>
<td>EGCM</td>
<td>Extended goal chasing method</td>
</tr>
<tr>
<td>GCM</td>
<td>Goal chasing method</td>
</tr>
<tr>
<td>JIT</td>
<td>Just-in-time</td>
</tr>
<tr>
<td>JITPS</td>
<td>Just-in-time production system</td>
</tr>
<tr>
<td>JITSP</td>
<td>Just-in-time sequencing problem</td>
</tr>
<tr>
<td>MA</td>
<td>Miltenburg’s algorithm</td>
</tr>
<tr>
<td>MA3H1</td>
<td>Miltenburg’s algorithm 3 with heuristic 1</td>
</tr>
<tr>
<td>MA3H2</td>
<td>Miltenburg’s algorithm 3 with heuristic 2</td>
</tr>
<tr>
<td>MBD</td>
<td>Mean-based divisor</td>
</tr>
<tr>
<td>MDJIT</td>
<td>Maximum deviation just-in-time</td>
</tr>
<tr>
<td>MEP</td>
<td>Method of equal proportion</td>
</tr>
<tr>
<td>NP-hard</td>
<td>Non-deterministic polynomial-time hard</td>
</tr>
<tr>
<td>ORVP</td>
<td>Output rate variation problem</td>
</tr>
<tr>
<td>PRVP</td>
<td>Product rate variation problem</td>
</tr>
<tr>
<td>RTVP</td>
<td>Response time variability problem</td>
</tr>
<tr>
<td>SCL</td>
<td>Supply chain logistics</td>
</tr>
<tr>
<td>SDC</td>
<td>Small deviation conjecture</td>
</tr>
<tr>
<td>SDJIT</td>
<td>Sum deviation just-in-time</td>
</tr>
<tr>
<td>TRSP</td>
<td>Truck sequencing problem</td>
</tr>
</tbody>
</table>
LIST OF PAPERS

The present thesis contains the following papers:


The present thesis contains the following papers:


**Paper E**: Thapa GB and Dhamala TN (2009) A synthetic study to minimize the inequality measures in JIT sequencing problem via optimiza-


**Paper J:** Thapa GB (2005) Elements of modern optimization, Epsilon-Delta (\(\epsilon - \delta\)) Vol. 2, pp. 49-60, Central Department of Mathematics, TU.


Contents

1 Introduction 1

2 Literature Surveys 7

2.1 Just-in-Time Production System . . . . . . . . . . . . . . . . 7

2.1.1 Pull Systems ....................... 10

2.1.2 Reduced Inventory .................... 11

2.1.3 Continuous Improvement ................. 12

2.2 Mixed-model JIT Sequencing Problem . . . . . . . . . . . . . 13

2.3 Discrete Apportionment Problem . . . . . . . . . . . . . . . . 15

3 Mathematical Models of JITSP 19

3.1 Product Rate Variation Problem . . . . . . . . . . . . . . . . 20

3.1.1 Mathematical Models with General Objectives . . . . . 20

3.1.2 Maximum and Sum Deviation Objectives . . . . . . . . . 26

3.1.3 Simultaneous Optimization . . . . . . . . . . . . . . . . 29

3.2 Output Rate Variation Problem . . . . . . . . . . . . . . . . . 30
Contents

1 Introduction 1

2 Literature Surveys 7
   2.1 Just-in-Time Production System .......................... 7
       2.1.1 Pull Systems ..................................... 10
       2.1.2 Reduced Inventory ................................. 11
       2.1.3 Continuous Improvement ............................. 12
   2.2 Mixed-model JIT Sequencing Problem ................... 13
   2.3 Discrete Apportionment Problem ......................... 15

3 Mathematical Models of JITSP 19
   3.1 Product Rate Variation Problem ........................ 20
       3.1.1 Mathematical Models with General Objectives ...... 20
       3.1.2 Maximum and Sum Deviation Objectives .......... 26
       3.1.3 Simultaneous Optimization ........................ 29
   3.2 Output Rate Variation Problem ........................ 30
4 Just-in-Time Sequencing Approaches

4.1 Heuristics for Product Rate Variation .......................... 40
  4.1.1 Miltenburg’s Heuristics .................................. 40
  4.1.2 Earliest Due Date Rule .................................. 42
  4.1.3 Two-stage Algorithm .................................. 43
  4.1.4 Final Assembly Sequencing Algorithm .................. 44
  4.1.5 Dynamic Programming Approach .................. 45

4.2 Heuristics for Output Rate Variation .......................... 48
  4.2.1 Goal Chasing Method I .................................. 49
  4.2.2 Goal Chasing Method II .................................. 50
  4.2.3 Extended Goal Chasing Method .................. 51
  4.2.4 The Pegging Assumption .................................. 51
  4.2.5 Dynamic Programming Solution .................. 52

4.3 Perfect Matching for Bottleneck PRVP .......................... 56
  4.3.1 Bounds for Bottleneck PRVP .......................... 59
  4.3.2 Improved Upper Bottleneck .......................... 61
  4.3.3 Bisection Search .................................. 62
  4.3.4 Problem with Two Products .......................... 63

5 Discrete Apportionment Approach: An Efficient Frontier 91

5.1 Formulation of Discrete Apportionment Problem .................. 92
5.2 JIT Sequencing versus Apportionment .......................... 93
5.3 On the Methods of Apportionment .......................... 94
5.4 Divisor Methods .......................... 96
  5.4.1 Parametric Methods .................................. 100
5.5 Mean-based Divisor Methods .......................... 101
5.6 Efficient Frontiers .......................... 103
  5.6.1 The Global Deviation Approach .................. 104
  5.6.2 The Local Deviation Approach .................. 106
5.7 Proportional Representation: A Counter Example .......................... 108
4.3.5  Problem with Small Bottlenecks ........................ 64
4.4  Bottleneck PRVP with General Index ........................ 68
4.5  Assignment Approach for Total PRVP ........................ 72
4.5.1  Feasibility and Optimality .............................. 77
4.5.2  Cyclic Sequences ...................................... 79
4.5.3  Weighted Bipartite Matching .............................. 82
4.6  Sequencing over Bicriterion Objectives ...................... 87

5  Discrete Apportionment Approach: An Efficient Frontier 91
5.1  Formulation of Discrete Apportionment Problem ............... 92
5.2  JIT Sequencing versus Apportionment ........................ 93
5.3  On the Methods of Apportionment ............................. 94
5.4  Divisor Methods .......................................... 96
5.4.1  Parametric Methods ................................... 100
5.5  Mean-based Divisor Methods ................................ 101
5.6  Efficient Frontiers ....................................... 103
5.6.1  The Global Deviation Approach .......................... 104
5.6.2  The Local Deviation Approach ............................ 106
5.7  Proportional Representation: A Counter Example ............. 108

6  PRVP Related Problems, Supply Chain and Queueing 113
6.1  PRVP Related Problems ...................................... 113
6.1.1  Response Time Variability Problem ....................... 114
CONTENTS

6.1.2 Liu-Layland Periodic Scheduling .......................... 116
6.1.3 Pinwheel Scheduling Problem .............................. 117
6.1.4 Fair Queueing ........................................... 119
6.1.5 Just-in-Time Delivery ...................................... 121
6.2 Supply Chain .................................................. 123
   6.2.1 Supply Chain Logistics ................................. 125
   6.2.2 Cross-Docking Operations ............................... 126
   6.2.3 The CDSCL Model Description ........................... 129
   6.2.4 The CDSCL Model Formulation ........................... 132
6.3 On Performance Modeling of Queueing ........................ 135

7 Some Applications ............................................. 139

8 Conclusion, Discussion and Further Works .......................... 147
List of Figures

2.1 Pull System in JITPS .......................... 10
2.2 Single-model, Mixed-model and Multi-model Lines .......................... 14
3.1 Actual and Ideal Schedules .......................... 22
3.2 Multi-level JITPS with Extended Supply Chain Network .......................... 31
4.1 Level curves for $d_1 = 4; B = 1$ to problem $F_{max}^a$ .......................... 69
6.1 Logistics networks in supply chain .......................... 124
6.2 Freight consolidation at the transfer terminal .......................... 127
6.3 Operational scenario at cross-docking center (CDC) .......................... 128
7.1 Information sharing model in supply chain .......................... 143
List of Tables

3.1 Cumulative production for $d_1 = 7, r_1 = 1$. .......... 2

3.2 Cumulative production for $d_2 = 5, r_2 = 14$. .......... 2

3.3 Cumulative production for $d_3 = 2, r_3 = 17$. .......... 2

4.1 Ideal positions for $d_1 = 3, d_2 = 4, d_3 = 5$; $D = 12$. .......... 2

4.2 The costs $\Psi_{ij}$ for $F_a \sum$ with $d_1 = 3, d_2 = 4, d_3 = 5$; $D = 12$. .......... 2

4.3 The additional costs $C_{ijk}$ for $F_a \sum$ with $d_1 = 3, d_2 = 4, d_3 = 5$; $D = 12$. .......... 2

5.1 Mutual Transformation: JIT vs. Apportionment .......... 94

5.2 The five divisor methods of apportionment .......... 98

5.3 Proportional Representation for 335 seats in CAE-08, Nepal .......... 111
List of Tables

3.1 Cumulative production for \( d_1 = 7, r_1 = \frac{1}{2} \) ........................ 24
3.2 Cumulative production for \( d_2 = 5, r_2 = \frac{5}{14} \) ..................... 25
3.3 Cumulative production for \( d_3 = 2, r_3 = \frac{1}{7} \) ........................ 26

4.1 Ideal positions for \( d_1 = 3, d_2 = 4, d_3 = 5; D = 12 \) .......... 74
4.2 The costs \( \Psi_{jl}^i \) for \( F_{sum}^a \) with \( d_1 = 3, d_2 = 4, d_3 = 5; D = 12 \) . . . 75
4.3 The additional costs \( C_{jk}^i \) for \( F_{sum}^a \) with \( d_1 = 3, d_2 = 4, d_3 = 5; D = 12 \) ........................................ 76

5.1 Mutual Transformation: JIT vs. Apportionment .......................... 94
5.2 The five divisor methods of apportionment ................................. 98
5.3 Proportional Representation for 335 seats in CAE-08, Nepal ........ 111
Chapter 1

Introduction

The central goal of this dissertation is to study and analyse some of the aspects of just-in-time (JIT) sequencing of products or jobs in mixed-model production system establishing an efficient frontier. The discrete apportionment characterization of the JIT sequencing problems via divisor methods, and the simultaneous study of production and supply chain logistics are further objectives of this dissertation. The basic concept of JIT sequencing is the ideal production and that of JIT logistics is ideal distribution. The synchronized research in production and logistics is being significant both in academic and industrial areas with real-world applications.

The aim of manufacturing industries is to produce goods that can be effectively sold to customers with maximum level of satisfaction and with minimum level of inventories. The raw materials, energy, capital, human resources and information are acquired, transported and consumed to complete the production cycle and supply chain. The manufacturing companies always aim for optimizing the resources consumed during this transformation by reducing the non-value added cost associated with overproduction, defects, inventory, transportation, waiting, motion, people and non-value added processing [109].
Chapter 1

Introduction

The central goal of this dissertation is to study and analyse some of the aspects of just-in-time (JIT) sequencing of products or jobs in mixed-model production system establishing an efficient frontier. The discrete apportionment characterization of the JIT sequencing problems via divisor methods, and the simultaneous study of production and supply chain logistics are further objectives of this dissertation. The basic concept of JIT sequencing is the ideal production and that of JIT logistics is ideal distribution. The synchronized research in production and logistics is being significant both in academic and industrial areas with real-world applications.

The aim of manufacturing industries is to produce goods that can be effectively sold to customers with maximum level of satisfaction and with minimum level of inventories. The raw materials, energy, capital, human resources and information are acquired, transported and consumed to complete the production cycle and supply chain. The manufacturing companies always aim for optimizing the resources consumed during this transformation by reducing the non-value added cost associated with overproduction, defects, inventory, transportation, waiting, motion, people and non-value added processing [109].
The just-in-time production system (JITPS) originated in Toyota company of Japan [64] has addressed the above issues and achieved a great success in automobile industries. The aim of JITPS is to produce only the needy products in demanded quantities at the right time in perfect quality. The JITPS has been used in mixed-model assembly lines with negligible change-over costs between the products to respond the customer demands for a variety of models of a common base product without holding large inventories or incurring large shortages. The important problem for the effective utilization of the system is the sequencing of different products keeping the usage rate of all parts used by the assembly lines as constant as possible. This is the mixed-model JIT sequencing problem (JITSP).

The sequencing of products determines the rate at which the raw materials are used to produce the goods in the respective product levels. This usage rate of materials is especially sensitive to the production sequence when several different products are to be produced on an assembly line. The time needed to switch from one product to another product, which is called the changeover time, is assumed to be negligible in this research. If the required changeover time between different products is not negligible, then there is another objective to minimize the amount of total changeover time and setup cost as well. Both of the objectives, minimizing material usage rates and setup times are considered simultaneously and Tabu search is used to find heuristic solution in [84].

The mixed-model JITSP minimizes the variations in demand rates of models produced on the assembly lines and the variations in demand ratios of outputs of supplying parts to produce models. The mixed-model JITSP that minimizes the rate at which different models are produced on the line is called the product rate variation problem (PRVP) [76]. The mixed-model JITSP that minimizes the demand ratios of supplying parts to produce models is called the output rate variation problem (ORVP) [76].
The just-in-time production system (JITPS) originated in Toyota company of Japan [64] has addressed the above issues and achieved a great success in automobile industries. The aim of JITPS is to produce only the needy products in demanded quantities at the right time in perfect quality. The JITPS has been used in mixed-model assembly lines with negligible change-over costs between the products to respond the customer demands for a variety of models of a common base product without holding large inventories or incurring large shortages. The important problem for the effective utilization of the system is the sequencing of different products keeping the usage rate of all parts used by the assembly lines as constant as possible. This is the mixed-model JIT sequencing problem (JITSP).

The sequencing of products determines the rate at which the raw materials are used to produce the goods in the respective product levels. This usage rate of materials is especially sensitive to the production sequence when several different products are to be produced on an assembly line. The time needed to switch from one product to another product, which is called the changeover time, is assumed to be negligible in this research. If the required changeover time between different products is not negligible, then there is another objective to minimize the amount of total changeover time and setup costs as well. Both of the objectives, minimizing material usage rates and setup times are considered simultaneously and Tabu search is used to find heuristic solution in [84].

The mixed-model JITSP minimizes the variations in demand rates of models produced on the assembly lines and the variations in demand ratios of outputs of supplying parts to produce models. The mixed-model JITSP that minimizes the rate at which different models are produced on the line is called the product rate variation problem (PRVP) [76]. The mixed-model JITSP that minimizes the demand ratios of supplying parts to produce models is called the output rate variation problem (ORVP) [76].

PRVP is the single-level problem and the ORVP is the multi-level problem. The PRVP and the ORVP have been formulated as non-linear integer programming problems with the objective of minimizing the deviations between the actual and the ideal productions under the assumption that the system has sufficient capacity with negligible switch-over costs from one model to another and each model is produced in a unit time [85, 88].

Most of the instances of PRVP are solved polynomially or pseudo-polynomially in many noble research works of various mathematicians and other scientists [71, 74, 76, 103, 104]. But it has been proved that ORVP is NP-hard [76] even though some sequencing heuristics and dynamic programming are devised for near optimal solutions, and the pegging assumption exists to reduce the problem into weighted PRVP.

Both the problems, PRVP and ORVP, are studied in perspective of the two types of objective functions, namely maximum deviation and sum deviation. The PRVP with the objective of minimizing the maximum deviation between the actual and the ideal productions is called the bottleneck PRVP. Similarly, the PRVP with the objective of minimizing the total deviations between the actual and the ideal productions is called the total PRVP. The total PRVP with a general objective function has been solved in a pseudo-polynomial time [71, 74].

The other problem that we study comparatively is the discrete apportionment problem (DAP), which is concerned with the fair allocation of congressional seats in integral form within the house of representatives. The rules of apportionment are vital elements to maintain every social and political order. In marriages and families, in business partnerships and social organizations, and in every government and supranational relationship, the rules of apportionment exist in various written and unwritten forms. In every form, the rule of apportionment affects not only how collective decisions are made, but also how and why a particular constitutional order develops over time. These
rules are important because the combination of their distributional and informational characteristics often incites particularly to contentious types of political conflict [6, 112].

In this dissertation, upon analysing the existing solution approaches to PRVP and ORVP, we have established a relation between total PRVP and discrete apportionment problem based on divisor methods. On formulating the two mean-based divisor methods, one of them being parametric [Section 5.5], we have proposed a stronger upper bound establishing the equitably efficient solution [Section 5.6]. On top of this, we have proved the necessary and sufficient conditions for the absolute and square deviation objectives of both total PRVP and DAP connecting with the related corollary for complexities. Similarly, we have shown that state to state variation problem and product to product rate variation problem are equivalent. Also, we have identified the tighter bound for bottleneck PRVP with absolute deviation objective [Subsection 4.3.2] proposing two conjectures. The new bound significantly reduces the deviation between actual and ideal productions. We have further pointed out the connection of production problem with the distribution problem providing a mathematical model to minimize the discrepancy of operation times of inbound and outbound trucks [Section 6.2]. Some PRVP related problems are explained and the cross-docking operations for supply chain logistics are presented followed by some applications of JITPS and supply chain.

**The outline of the succeeding Chapters is as follows:**

We begin with the brief literature surveys of JITPS and discrete apportionment problem in Chapter 2 together with their fundamental characteristics. Further we explain some related terminologies such as pull systems, reduced inventory, continous improvement and mixed-model production systems. The mathematical model formulations of PRVP and ORVP are presented in Chapter 3 with corresponding particular objective functions fol-
In Chapter 4, some sequencing heuristics and the other solution strategies for the JITSPs are extensively reported, namely heuristic approaches for PRVP and ORVP including dynamic programmings for each, perfect matching approach for the bottleneck PRVP and assignment formulation for the total PRVP are analysed, pegging assumption is described for ORVP to reduce it into weighted PRVP; and sequencing over bicriterion objectives are also described with Pareto algorithm. Moreover, we have investigated a tighter upper bound for bottleneck PRVP with two conjectures.

Chapter 5 consists of an efficient frontier for total PRVP via discrete apportionment methods. Furthermore, mean-based divisor methods are explored, equitably efficient solution is established, and the problem is handled with global and local deviation approaches. Chapter 6 includes some PRVP related problems and supply chain logistics as the linkage of production problem with the distribution problem. The cross-docking logistics operations are explained and the cross-docking supply chain logistics model is formulated to minimize the deviation of operation times of outbound and inbound trucks. Furthermore, the performance modeling of queueing system is briefly described as queueing is also related to sequencing of jobs, products and services in production assembly lines. Chapter 7 presents a short note on some of the real-life applications of JITPS and supply chain logistics. And, finally Chapter 8 concludes the dissertation opening the floor for further research.
Chapter 2 is based on the papers A, H, I, J. In this chapter, the literature surveys of just-in-time production system and discrete apportionment problem are presented in brief. The simultaneous dealing of the just-in-time sequencing problem and the discrete apportionment problem is given in Chapter 5 together with their mathematical interlink. Here, Section 2.1 presents the just-in-time production system with short historical note. Mixed-model just-in-time sequencing problem is described in Section 2.2, and the discrete apportionment problem is explained in Section 2.3 with its basic properties and paradoxes.

2.1 Just-in-Time Production System

Production system has been a human activity for a very long time in this or that form, for instance, the ancient cavemen started the Stone Age by producing the stone articles and the wooden spears for their defensive livelihood. Some of the developed production systems have been in practice since seventeenth century, such as large scale production by power-driven machinery focusing on new technology but without any concern of linkages for production.
Chapter 2

Literature Surveys

Chapter 2 is based on the papers A, H, I, J.

In this chapter, the literature surveys of just-in-time production system and discrete apportionment problem are presented in brief. The simultaneous dealing of the just-in-time sequencing problem and the discrete apportionment problem is given in Chapter 5 together with their mathematical interlink. Here, Section 2.1 presents the just-in-time production system with short historical note. Mixed-model just-in-time sequencing problem is described in Section 2.2, and the discrete apportionment problem is explained in Section 2.3 with its basic properties and paradoxes.

2.1 Just-in-Time Production System

Production system has been a human activity for a very long time in this or that form, for instance, the ancient cavemen started the Stone Age by producing the stone articles and the wooden spears for their defensive livelihood. Some of the developed production systems have been in practice since seventeenth century, such as large scale production by power-driven machinery focusing on new technology but without any concern of linkages for produc-
tion process. Gradually, several scientific methods for production systems have been proposed to address the variable demands of customers in optimal way, such as Toyota production system (TPS) in Japan, having its origin in Ford Motor Company in America.

The Ford Motor Company is credited in the development of just-in-time (JIT) notion, as described in Henry Ford’s book *My Life and Work* (1922), who applied scientific management proposed by F. W. Taylor, on a grand scale in the production of automobiles. It is noteworthy that production systems are dynamic, that is, an advanced manufacturing system at present may not necessarily be advanced in future. Thus, the very first cornerstone of just-in-time production system (JITPS) can be traced out from Ford production system (Ford Motor Company, 1903) where large assembly tasks were broken down to smaller tasks and products were assembled and fabricated station to station with distinct tasks carried out at each station. This system used notable strategies, such as globalization and optimization of supply base, long-term contracts, single sourcing and early supplier involvement.

Since the time of Ford, product requirements and hence the requirements of production systems have been changing rapidly. Assembly lines were originally developed for a cost efficient mass production of a single standardized product. Nowadays, varieties of options are available to the customers, so that manufacturers need to handle product varieties which exceed several billions of models. Though JITPS can be traced out from Ford production era around 1900s [111], the present idea of the system is developed and perfected by T. Ohno, while working as an assembly manager in Toyota motor company around 1970s, which revolutionized the Toyota production system (TPS) to maintain the stable production system by eliminating waste and by autonomination. The basic idea of TPS is to maintain a continuous flow of raw materials and final products in the factories to adapt the changes in demand with flexibility.
Standardized work, smoothing production schedule via mixed-model sequencing and the change for better are the main bases of TPS. The JITPS is a management philosophy based on the planned elimination of all wastages, continuous improvement of productivity and reduction of inventories in all level; performed by producing only the necessary amount of necessary products in perfect quality at right place and time [89, 112]. To achieve this goal, the JITPS penalizes the early-tardy jobs by using the limited resources (e.g., manpower, materials, machinery, space and time) in optimal way. The main target is to satisfy customers for various demands of different products without holding large inventories and without incurring large shortages of products.

The key features of TPS (i.e., JITPS) are low inventories, stable and level production rates, reduction of lot sizes and pull system. Any manufacturing company using the JIT approach must integrate all of these features to function successfully. The major benefits arising from the use of JITPS are reduced inventory levels of raw materials, work-in-process and finished goods; increased product quality and a reduction of scrap and rework; a reduction in lead times and a greater flexibility in changing the production mix; a smoother flow of production with shorter set-up times, multi-skilled workers and fewer disruptions due to quality problems; reduced space requirements due to an efficient plant layout and lower inventory levels [42]. Since the focus in JIT manufacturing is on solving production problems, the manufacturing operations are being increasingly more streamlined and problem-free.

The JITPS is a manufacturing system that attempts to produce with the shortest possible lead-time, with the lowest possible inventory and with the fewest possible waste focusing to achieve excellence through the principles of continuous improvement and waste reduction. Moreover, JITPS is pull system where products are assembled just before they are sold, subassemblies are made just before the products are assembled and components are fabricated just before the subassemblies are made [112].
2.1.1 Pull Systems

A facility operating under JITPS uses a pull system that pulls the products according to their demands. In this system, work is moved from operation to operation only in response to demand from the next stage in the process. The control of this movement is the responsibility of the subsequent operation. Each workstation pulls the output from the previous station only when it is needed. Output of the finished goods for the entire production facility is pulled by customer demand (See Figure 2.1). Communication occurs backward through the system from station to station. Work moves just-in-time for the next operation and the flow of work is coordinated in such a way that the accumulation of excessive inventory between operations is avoided. We refer \[98\] for a detail description of push and pull systems.

![Figure 2.1: Pull System in JITPS](image)

Due to this pull nature, the JITPS differs from traditional push production system which pushes materials to the next stage of the production without coping with demand and time needed at the next level of production. This
creates lots of inventories at each level of the production flow. To skip this situation, the JITPS is based on the concept of pull production which eliminates the total inventory. The information of the demand in this system can be achieved in a variety of ways. The most commonly used device is some variant of the kanban card system used at Toyota (the terms JITPS and kanban production systems are often interchangeable). When materials or work are required from the preceding station, a kanban card is sent authorizing the move or work for parts. No part or lot can be moved or worked on without the use of these cards. The use of kanban cards at Toyota to control their JIT process is described in [90].

Thus, the pull system responds to real-world demands or orders and forces the upstream process to respond, whereas a traditional push system uses a schedule based on prediction of demand. The core difference between a pull system and a push system is the process trigger. The important technical elements for pull systems to succeed are: flowing product in small batches (approaching one piece flow where possible), pacing the processes to takt time (to stop overproduction), signaling replenishment via a kanban signal, and leveling of product mix and quantity over time. The scheduling department must set the right mix and quantity of products to be produced in a sequence. This can be done by placing production kanban cards in a heijunka box (a visual scheduling tool used to achieve a smoother production flow), often at the beginning of each shift. A sequential system requires strong management to maintain the overall production systems, and hence improving it may be a challenge.

2.1.2 Reduced Inventory

The most promising feature of JITPS is the reduced inventory in all levels of production for saving both space and resources. Production problems that might be hidden in the inventory of a traditional manufacturing environment
are exposed in the JITPS and may be corrected in the evolutionary approach taken in problem solving [61]. High level of inventory creates problems such as, high setup time, quality defects, equipment down time, production planning deficiencies and so on. There should be a continuous drive from top to bottom to minimize all types of inventories: work-in-process, raw materials, component parts, finished goods and so on.

A significant amount of inventory can be reduced by the demand predictions at different points of production flow. JIT techniques can be used to lessen the inventories, for instance- electronic data interchanges (EDI). The reduction of inventory increases the productivity by increasing the volume of production from given resources lowering the cost of product. Productivity is the ratio of output to the value of input. The outputs may be products or services and the inputs or resources may be land, materials, plant machineries, tools and a human resource. As a result of improved productivity, reduced inventory and continuous improvement, the JITPS is an efficient and effective production system including three sequential components: people involvement → total quality control → JIT flow; jointly called productivity triad [112].

2.1.3 Continuous Improvement

The JITPS is based on a good team work involving suppliers, workers and customers. In this system, the companies adopt continuous improvement in quality and productivity by identifying areas that require further improvement. Problems are detected before they occur and solved in the minimum possible time to ensure smooth flow of work. The percentage of scrap in the manufacturing operations can be reduced by following better work methods and by giving suitable trainings to the employees. A proper master production schedule and flexible workforce can be developed to eliminate the unnecessary shortages and inventories.
The manufacturing operations can be improved on a continuous basis by the complete involvement of employees and the management. For service operations, the process of continuous improvement targets to reduce the number of people involved in providing the service without affecting the quality and rate of service. The JIT system is a powerful tool for reducing the firms inventory and improving productivity. However, implementation of JIT principles is a difficult task, for instance workers motive to change, difficulty in accomplishing zero lead time, zero safety stock, and zero idle time have to be overcome. To overcome these defects, JITPS adopts economic setups and produces in small lot sizes which has shorter waiting time in the production process. Support and commitment from the top management and extensive employee training lead to the successful implementation of JITPS.

2.2 Mixed-model JIT Sequencing Problem

Most of the manufacturing companies today use mixed-model production systems for diversified small-lot production to address the variable demands of different customers. Mixed-model sequencing is used when a variety of products with similar nature are produced alternately on the same production line. Differences in the products may vary in small ways such as the colour of paint on a component, or they may vary greatly, such as a base model versus a luxury package, or even front wheel drive and rear wheel drive automobiles. In mixed-model production system, the manufacturing firms intentionally interplay in mixing the products, balancing the scheduling and smoothing the workload. This method of manufacturing replaces the well-known large batch production method.

Sequencing problem is a problem that finds a permutation or an ordering of a finite collection of jobs or products that satisfies certain conditions, such as precedence constraints, integrality constraints, monotonic constraints and
etc [61]. In JIT sequencing (scheduling) environment, products (jobs) that complete early must be held in finished goods inventory till their due dates, while products that complete after their due dates may cause customers to shut down operations. So an ideal schedule is one in which all products are finished exactly on their assigned due dates.

The concept of penalizing both earliness and tardiness has spawned a new and rapidly developing line of research in scheduling theory. The mixed-model JIT sequencing is the problem of determining production sequence of different models of the same product produced on the assembly line, assuming
that products require an approximately equal number and mix of parts. The Figure 2.2 best pictorizes the three types of production systems, namely single-model lines, mixed-model lines and multi-model lines respectively.

2.3 Discrete Apportionment Problem

There is a very large class of real life problems related to fair division of resources among competing interests in many areas of applications, which plays a significant role in decision sciences. Several types of equity problems arise in allocating available resources in integral parts to different subdivisions. Some of the problems are efficiently solved (e.g., assignment problem) whereas others are not well-solved yet (e.g., timetabling problem). One particular problem having wide applications in governmental decision-making is the apportionment problem, which may be of continuous-type, for example apportionment of taxes and of discrete-type, for example apportionment of seats in house of representatives.

The discrete apportionment problem plays an important role in modern democracies, a classical example being the U.S. presidential election, which is the problem of translating an election outcome to a number of seats in fixed-size political house. Mathematically, it is the problem of translating a sequence of reals to a sequence of integers, while ensuring that the sum of the sequence sums up to a pre-determined number, called the house size. The problem arises because seats are indivisible (integers), whereas an election outcome generally gives rise to fractional remainders (reals). The main problem is to minimize the difference between these two quantities as close as possible.

The discrete apportionment problem (DAP) has its origin in the proportional election system developed for house of representatives in US, where each state receives seats in the house in proportion to its population [6]. DAP occurs in
all kinds of electoral systems, for example, in the following political systems:

(a) **Federal system**: it is regional representation system based on population, for example in USA.

(b) **Proportional system**: it is political representation system based on votes, for example in Israel.

(c) **Mixed system**: it is mixture of federal and proportional systems, for example in Nepal.

The DAP is the problem of determining how to divide a given integer number of representatives or delegates proportionally among the given constituencies according to their respective sizes. It is a quite complex kind of discrete fair-division problem in electoral systems, because all possible apportionment methods contradict the principle of fairness criteria [23]. In fact, no method equalizes states under the fixed house size allocating minimum requirement of one seat and states not crossing the house size. Mainly two fairness ideas have been studied in the apportionment literature: the first is each state should get either its lower quota or upper quota and the second is to look at pairwise equity between states.

There always exists a certain inequality between two states yielding one of the states a slight advantage over the other. A transfer of one seat from the more favored state to the less favored state will ordinarily reverse the sign of inequality, so that the more favored state now becomes the less favored, and vice-versa. Whether such a transfer should be made or not, depends on whether the amount of inequality between the two states after the transfer, is less or greater than it was before. If the amount of inequality is reduced by the transfer, then it is obvious that the transfer should be made. The fundamental problem of quite unexpected complexity is, therefore, how to measure the amount of inequality between two states and how to minimize it as far as possible, since it cannot be eliminated perfectly. The philosophy of apportionment must obey political legitimacy and the solutions must be acceptable to nation [6].
It naturally appears that finding a perfect apportionment method is a difficult job. Some methods of apportionment are practiced in different time and situations. The first and simplest method is the largest reminder methods proposed by A. Hamilton and known as Hamilton methods which was modified by others as well. The next family of methods are the divisor methods developed by E. V. Huntington based on a wise notion of rounding. Similarly parametric divisor methods, quota methods, balanced methods are also in existence in the apportionment literature [32].

For an excellent historical note, mathematical formulation and the apportionment methods, we refer the seminal monograph by Balinski and Young [6]. We formulate the DAP in Section 5.1. Though there is not a single method of apportionment meeting all the requirements imposed by political needs, a so-called perfect apportionment method is supposed to satisfy some basic properties [101, 112]: (a) Quota condition: each state should have its seats within the lower quota and the upper quota, (b) House monotonicity: when total number of representatives (house size) increases, then any state’s number of representatives should not decrease, (c) Population monotonicity: the number of representatives of any state should not decrease as its population increases. Also, any method should not artificially favor large states at the expense of smaller ones and vice-versa, (d) Quota monotonicity: the actual apportionment of any state should not decrease as its quota increases, (e) Minimum requirement: every state must have at least one representative.

As a result of using one method or another, some surprising apportionment paradoxes are found in the apportionment practices [112]: (a) Alabama paradox: an increase in the size of the house can cause a state to lose a seat. Hamilton method in 1880 assigned Alabama state 8 seats from house size 299, whereas it gave only 7 seats from increased house size 300, (b) Population paradox: an increase in a state’s population can cause it to lose a seat faced around 1900 in Hamilton method. If the population of one state is increased while holding the other state’s population and house size fixed,
the former state may lose a seat [106], (c) New states paradox: adding new state and increasing house size can cause another state to lose seats, found in 1907 when Oklahoma became a new state. As a new state, it got 5 new seats increasing the old house size from 386 to 391. As a result, Maine’s apportionment went up from 3 to 4 and New York’s went down from 38 to 37. But the intent was to leave unchanged for other states, (d) Quota paradox: a state may receive number of seats less than its lower quota or more than its upper quota; faced while applying Jefferson’s method, (e) Migration paradox: under the fixed total population, house size, number of states and fixed population of a state, it is possible that the state can lose seats if there is a population shift between other two states. The migration paradox affects both Hamilton and divisor methods. These paradoxes are visualized via geometric representation in [22].
Chapter 3

Mathematical Models of JITSP

Chapter 3 is based on the papers A, E, F.

The first attempt in solving the real-world problem is to model it in mathematical form. The just-in-time sequencing problem (JITSP) has been modeled in single-level case and multi-level case under some constraints as nonlinear integer programming problems. The JITSP in single-level case is referred as product rate variation problem (PRVP) and the JITSP in multi-level case is referred as output rate variation problem (ORVP).

In this chapter, we give the mathematical formulations for PRVP and ORVP with their respective particular cases. The PRVP is formulated with general objectives considering convex symmetric penalty function in Section 3.1 describing the maximum deviation and the sum deviation objectives followed by the brief discussion of simultaneous optimization. Similarly, the ORVP as well as its particular cases are formulated in Section 3.2 describing NP-hard results.
3.1 Product Rate Variation Problem

The product rate variation problem (PRVP) is the mixed-model single-level JIT sequencing problem. In the following two Subsections, we present the mathematical models for the PRVP in the maximum deviation case and the total deviation case.

3.1.1 Mathematical Models with General Objectives

Suppose there are $n$ products to be produced in the given span of time period 1 through $k$ with the integer demands $d_1, d_2, \ldots, d_n$ such that $\sum_{i=1}^{n} d_i = D$, the total demand. The time needed to produce one unit is assumed to be independent on the product and time needed to switch from one product to another is assumed to be negligible. Without loss of generality, it can be supposed that it takes one unit of time to produce one unit of product and thus the time horizon is equal to $D$ time units. If $r_i = \frac{d_i}{D}$ is the ideal production rate for the parts of type $i$ such that $\sum_{i=1}^{n} r_i = 1$, then the scheduling goal for the assembly line is to maintain the total cumulative production of product $i$ to the total production as close to $r_i$ as possible. This means exactly $kr_i$ units of product $i$ should be produced in the first $k$ time periods ($k = 1, 2, \ldots, D$), which is the ideal production.

Let $x_{ik}, i = 1, 2, \ldots, n; k = 1, 2, \ldots, D$, be the actual cumulative production of product $i$ in the time period 1 through $k$ and $kr_i$ be the ideal production of the product $i$ in the same time horizon. Clearly, $x_{ik} - kr_i > 0$ implies inventories and $kr_i - x_{ik} > 0$ implies shortage of the products, and $kr_i - x_{ik} = 0$ implies the ideal production (See Figure 3.1). For a convex symmetric penalty function $F_i, i = 1, 2, \ldots, n$ with minimum $F_i(0) = 0$; the maximum deviation just-in-time (MDJIT) and the sum deviation just-in-time (SDJIT)
sequencing problems are formulated as follows [31, 33, 112]:

\[
F_{\text{max}} = \min_{i,k} \max \sum_{i=1}^{n} F_i(x_{ik} - kr_i) \tag{3.1}
\]

\[
F_{\text{sum}} = \min \sum_{k=1}^{D} \sum_{i=1}^{n} F_i(x_{ik} - kr_i) \tag{3.2}
\]

subject to

\[
\sum_{i=1}^{n} x_{ik} = k, \quad k = 1, 2, \ldots, D \tag{3.3}
\]

\[
x_{i(k-1)} \leq x_{ik}, \quad i = 1, 2, \ldots, n; \quad k = 2, 3, \ldots, D \tag{3.4}
\]

\[
x_{iD} = d_i, x_{i0} = 0, \quad i = 1, 2, \ldots, n \tag{3.5}
\]

\[
x_{ik} \geq 0, \text{ integer}, \quad i = 1, 2, \ldots, n; \quad k = 1, 2, \ldots, D \tag{3.6}
\]

The constraint (3.3) ensures that exactly $k$ units of various models $i, i = 1, 2, \ldots, n$ are produced during the periods 1 through $k$. The constraint (3.4) represents the monotone condition ensuring that the total production of every product is a non-decreasing function of $k$. The constraint (3.5) guarantees that the demands are exactly met for each product (obviously satisfied by any optimal solution) whereas the constraint (3.6) is the integrality constraint. These four constraints jointly indicate that exactly one product is produced during each stage. Moreover, the above formulation (3.1) to (3.6) is an integer programming problem with cardinality, monotonicity and integrality constraints. The optimization problem is to find the sequence $f = s_1, s_2, \ldots, s_D$ that minimizes one of the objectives (3.1) or (3.2) under the constraints (3.3) to (3.6). The deviations between the actual production $x_{ik}$ and ideal production $kr_i$ are sketched in Figure 3.1 as actual and ideal schedules.

The problem with the objective function $F_{\text{max}}$ under the constraints (3.3) – (3.6) is called the bottleneck product rate variation problem (PRVP), which minimizes the bottleneck measure of deviation that produces smooth sequence in every time unit [103]. The problem with the objective function
$F_{sum}$ under the constraints (3.3) – (3.6) is called the total product rate variation problem (PRVP) which minimizes the total measure of deviations that produces smooth sequence on the average [76].

The PRVP is sequencing of products in mixed-model assembly line to find a balanced sequence of different models of the same product, where each model is distributed as evenly as possible satisfying the demands for different models. This intermixing of the models is explained by an example below:

**Example:** Suppose there are three different models of a product to be produced. If the product is car, then the required models to be produced in mixed-model sequence may be a *red car* with manual transmission, a *blue car* with automatic transmission, and a *white car* with both systems. These

![Figure 3.1: Actual and Ideal Schedules](image-url)
models are represented as model A, model B, and model C; in short A, B, C respectively, and each model has different demands. Suppose the demands of models A, B and C be $d_1 = 7$, $d_2 = 5$ and $d_3 = 2$ units respectively such that the total demand is $D = 14$. One of the sequences based on batch production system looks like:

$$\text{Sequence } l: \text{ A A A A A B B B C C}$$

In this system, the number of setups is minimized. But production is unsynchronized with market demand which generates large inventories and customers have to wait to get their choices. To avoid this situation, we have to intermix the models in the production line to exploit the benefits of mixed-model sequencing. Some of the possible mixed-model sequences are:

- Sequence 1: A B A B A B C A B A B A C A
- Sequence 2: B A B A B A C B A B A A C A
- Sequence 3: C C A A B A B A B A B A B A
- Sequence 4: A B A A B A C B A B B C A A

Obviously there are several ways to find such sequences. The number of possible sequences associated with this example are

$$\frac{D!}{d_1!d_2!d_3!} = \frac{14!}{7!5!2!} = 72,072.$$
ideal productions). The position of each model in this production sequence is as below where model A is produced in the first position, model B is produced in the second, again model A in the third position and so on upto the fourteenth position:

Sequence: A B A A B A C B A B C A A
Position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Table 3.1: Cumulative production for $d_1 = 7, r_1 = \frac{1}{2}$.

<table>
<thead>
<tr>
<th>Time $(k)$</th>
<th>Actual production $(x_{ik})$</th>
<th>Ideal production $(kr_i)$</th>
<th>Deviations $(x_{ik} - kr_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x_{11} = 1$</td>
<td>$1r_1 = \frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>2</td>
<td>$x_{12} = 1$</td>
<td>$2r_1 = 1$</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$x_{13} = 2$</td>
<td>$3r_1 = \frac{3}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>4</td>
<td>$x_{14} = 3$</td>
<td>$4r_1 = 2$</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>$x_{15} = 3$</td>
<td>$5r_1 = \frac{5}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>6</td>
<td>$x_{16} = 4$</td>
<td>$6r_1 = 3$</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>$x_{17} = 4$</td>
<td>$7r_1 = \frac{7}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>8</td>
<td>$x_{18} = 4$</td>
<td>$8r_1 = 4$</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>$x_{19} = 5$</td>
<td>$9r_1 = \frac{9}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>10</td>
<td>$x_{110} = 5$</td>
<td>$10r_1 = 5$</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>$x_{111} = 5$</td>
<td>$11r_1 = \frac{11}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>12</td>
<td>$x_{112} = 5$</td>
<td>$12r_1 = 6$</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>$x_{113} = 6$</td>
<td>$13r_1 = \frac{13}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>14</td>
<td>$x_{114} = 7$</td>
<td>$14r_1 = 7$</td>
<td>0</td>
</tr>
</tbody>
</table>

We formalize the example with the above data in mathematical terms as follows [35]: Let $i, (i = 1, 2, 3)$ be the three models of Toyota car, $d_i, (d_1 = 7, d_2 = 5, d_3 = 2)$ be the demands for each model $i$ such that the total demand is $D = 14$, and $k, (k = 1, 2, \ldots, 14)$ be the time unit. We denote ideal production rate by $r_i = \frac{d_i}{D}$ such that $r_1 = \frac{1}{2}, r_2 = \frac{5}{14}, r_3 = \frac{1}{7}$, the ideal
ideal production rate by $r_D$ in the fourteenth position: produced in the second, again model A in the third position and so on up to the is as below where model A is produced in the first position, model B is pro-

ideal productions). The position of each model in this production sequence

follows \[35\]: Let \(d_k, r_k, \ldots, r_2 = 5\) be the demands for each model
of the three models of Toyota car, \(d_k, r_k, \ldots, r_2 = 2\) be the time unit. We denote \(kr_1 = 1\), \(kr_2 = 1\), \(kr_3 = 1\), \(kr_4 = 2\), \(kr_5 = 3\), \(kr_6 = 2\), \(kr_7 = 2\), \(kr_8 = 3\), \(kr_9 = 3\), \(kr_{10} = 4\), \(kr_{11} = 5\), \(kr_{12} = 5\), \(kr_{13} = 5\), \(kr_{14} = 5\), \(kr_{15} = 5\), \(kr_{16} = 3\), \(kr_{17} = 3\), and so on yielding the monotonicity constraint: $x_{i(k-1)} \leq x_{ik}$. It is clear that $x_{11} = 1$, $x_{12} = 1$, $x_{13} = 2$, $x_{14} = 3$ and so on generating the cardinality constraint: $\sum_{i=1}^{n} x_{ik} = k$. It is visible that $x_{10} = 0$ and $x_{114} = 7$ showing that the demands must be met by total production of each model: $x_{iD} = d_i$. Finally, the actual production must be integer: $x_{ik} \geq 0$, for $i = 1, 2, 3$; $k = 1, 2, \ldots, 14$.

<table>
<thead>
<tr>
<th>Time $(k)$</th>
<th>Actual production $(x_{ik})$</th>
<th>Ideal production $(kr_i)$</th>
<th>Deviations $(x_{ik} - kr_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x_{21} = 0$</td>
<td>1$r_2 = \frac{5}{14}$</td>
<td>$\frac{5}{14}$</td>
</tr>
<tr>
<td>2</td>
<td>$x_{22} = 1$</td>
<td>2$r_2 = \frac{5}{7}$</td>
<td>$\frac{2}{7}$</td>
</tr>
<tr>
<td>3</td>
<td>$x_{23} = 1$</td>
<td>3$r_2 = \frac{15}{14}$</td>
<td>$\frac{1}{14}$</td>
</tr>
<tr>
<td>4</td>
<td>$x_{24} = 1$</td>
<td>4$r_2 = \frac{10}{7}$</td>
<td>$\frac{3}{7}$</td>
</tr>
<tr>
<td>5</td>
<td>$x_{25} = 2$</td>
<td>5$r_2 = \frac{25}{14}$</td>
<td>$\frac{3}{14}$</td>
</tr>
<tr>
<td>6</td>
<td>$x_{26} = 2$</td>
<td>6$r_2 = \frac{15}{7}$</td>
<td>$\frac{1}{7}$</td>
</tr>
<tr>
<td>7</td>
<td>$x_{27} = 2$</td>
<td>7$r_2 = \frac{5}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>8</td>
<td>$x_{28} = 3$</td>
<td>8$r_2 = \frac{20}{7}$</td>
<td>$\frac{1}{7}$</td>
</tr>
<tr>
<td>9</td>
<td>$x_{29} = 3$</td>
<td>9$r_2 = \frac{45}{14}$</td>
<td>$\frac{3}{14}$</td>
</tr>
<tr>
<td>10</td>
<td>$x_{210} = 4$</td>
<td>10$r_2 = \frac{25}{7}$</td>
<td>$\frac{3}{7}$</td>
</tr>
<tr>
<td>11</td>
<td>$x_{211} = 5$</td>
<td>11$r_2 = \frac{55}{14}$</td>
<td>$\frac{15}{14}$</td>
</tr>
<tr>
<td>12</td>
<td>$x_{212} = 5$</td>
<td>12$r_2 = \frac{30}{7}$</td>
<td>$\frac{5}{7}$</td>
</tr>
<tr>
<td>13</td>
<td>$x_{213} = 5$</td>
<td>13$r_2 = \frac{65}{14}$</td>
<td>$\frac{5}{14}$</td>
</tr>
<tr>
<td>14</td>
<td>$x_{214} = 5$</td>
<td>14$r_2 = 5$</td>
<td>$0$</td>
</tr>
</tbody>
</table>

Table 3.2: Cumulative production for $d_2 = 5, r_2 = \frac{5}{14}$.}

production of product $i$ in time unit $k$ by $kr_i$ and the actual production of the same by $x_{ik}$.

From the calculations of $x_{ik}$ and $kr_i$ in the three tables, we notice that $x_{11} + x_{21} + x_{31} = 1; x_{12} + x_{22} + x_{32} = 1 + 1 + 0 = 2; x_{13} + x_{23} + x_{33} = 2 + 1 + 0 = 3$ and so on.
Table 3.3: Cumulative production for \( d_3 = 2, r_3 = \frac{1}{7} \).

<table>
<thead>
<tr>
<th>Time ((k))</th>
<th>Actual production ((x_{ik}))</th>
<th>Ideal production ((kr_i))</th>
<th>Deviations ((x_{ik} - kr_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x_{31} = 0 )</td>
<td>( 1r_3 = \frac{1}{7} )</td>
<td>( \frac{1}{7} )</td>
</tr>
<tr>
<td>2</td>
<td>( x_{32} = 0 )</td>
<td>( 2r_3 = \frac{2}{7} )</td>
<td>( \frac{2}{7} )</td>
</tr>
<tr>
<td>3</td>
<td>( x_{33} = 0 )</td>
<td>( 3r_3 = \frac{3}{7} )</td>
<td>( \frac{3}{7} )</td>
</tr>
<tr>
<td>4</td>
<td>( x_{34} = 0 )</td>
<td>( 4r_3 = \frac{4}{7} )</td>
<td>( \frac{4}{7} )</td>
</tr>
<tr>
<td>5</td>
<td>( x_{35} = 0 )</td>
<td>( 5r_3 = \frac{5}{7} )</td>
<td>( \frac{5}{7} )</td>
</tr>
<tr>
<td>6</td>
<td>( x_{36} = 0 )</td>
<td>( 6r_3 = \frac{6}{7} )</td>
<td>( \frac{6}{7} )</td>
</tr>
<tr>
<td>7</td>
<td>( x_{37} = 1 )</td>
<td>( 7r_3 = 1 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>8</td>
<td>( x_{38} = 1 )</td>
<td>( 8r_3 = \frac{8}{7} )</td>
<td>( \frac{1}{7} )</td>
</tr>
<tr>
<td>9</td>
<td>( x_{39} = 1 )</td>
<td>( 9r_3 = \frac{9}{7} )</td>
<td>( \frac{2}{7} )</td>
</tr>
<tr>
<td>10</td>
<td>( x_{310} = 1 )</td>
<td>( 10r_3 = \frac{10}{7} )</td>
<td>( \frac{3}{7} )</td>
</tr>
<tr>
<td>11</td>
<td>( x_{311} = 1 )</td>
<td>( 11r_3 = \frac{11}{7} )</td>
<td>( \frac{4}{7} )</td>
</tr>
<tr>
<td>12</td>
<td>( x_{312} = 2 )</td>
<td>( 12r_3 = \frac{12}{7} )</td>
<td>( \frac{2}{7} )</td>
</tr>
<tr>
<td>13</td>
<td>( x_{313} = 2 )</td>
<td>( 13r_3 = \frac{13}{7} )</td>
<td>( \frac{1}{7} )</td>
</tr>
<tr>
<td>14</td>
<td>( x_{314} = 2 )</td>
<td>( 14r_3 = 2 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

### 3.1.2 Maximum and Sum Deviation Objectives

The general sequencing objective functions (3.1) and (3.2) are studied as square and absolute deviation objectives under the same constraints in various research works [31, 33, 85, 112]. The various forms of discrepancy functions are presented in [111] as the minimization of inequality measures correlating with discrete apportionment problem. In the following, we present the maximum deviation and sum deviation objectives studied so far.

The MDJIT sequencing problems \(i.e.,\) bottleneck PRVP as absolute deviation and square deviation objectives are

\[
F_{max}^a = \min \max_{i,k} |x_{ik} - kr_i| \quad (3.7)
\]

\[
F_{max}^s = \min \max_{i,k} (x_{ik} - kr_i)^2 \quad (3.8)
\]
where \( a \) and \( s \) over \( F_{max} \) stand for absolute and square respectively.

The mostly studied SDJIT sequencing problems (i.e., total PRVP) as absolute deviation and square deviation objectives are [85]

\[
F^{a}_{sum} = \min \sum_{k=1}^{D} \sum_{i=1}^{n} \left| \frac{x_{ik}}{k} - r_{i} \right| \tag{3.9}
\]

\[
F^{s}_{sum} = \min \sum_{k=1}^{D} \sum_{i=1}^{n} \left( \frac{x_{ik}}{k} - r_{i} \right)^{2} \tag{3.10}
\]

Miltenburg [85] further deduced that

\[
F^{a'}_{sum} = \min \sum_{k=1}^{D} \sum_{i=1}^{n} |(x_{ik} - kr_{i})| \tag{3.11}
\]

\[
F^{s'}_{sum} = \min \sum_{k=1}^{D} \sum_{i=1}^{n} (x_{ik} - kr_{i})^{2} \tag{3.12}
\]

The tractability of either type of these objectives is equivalent mathematically. The discrepancy functions (3.9) and (3.10) aim to keep the actual proportions of the production mix \( \frac{x_{ik}}{k} \) close to the desired proportions \( r_{i} \) at all times \( k \), whereas (3.11) and (3.12) attempt to keep the actual number of units produced \( x_{ik} \) close to the desired number of units \( kr_{i} \) at all times. Both types of objectives yield reasonably similar schedules. Miltenburg [85] proposed three algorithms with two supporting heuristics for good solution of the objective (3.12), which is discussed in Chapter 4.

Defining the ideal production time \( t_{ik} = \frac{2k-1}{2r_{i}}, i = 1, 2, \ldots, n; k = 1, 2, \ldots, D \) and the needed production time \( y_{ik} \) of each product, Inman and Bulfin [56] proposed the following min-sum squared sequencing objective to be minimized

\[
f(y) = \sum_{i=1}^{n} \sum_{k=1}^{D} (y_{ik} - t_{ik})^{2} \tag{3.13}
\]
and developed a pseudo polynomial heuristic with complexity \(O(D)\) via an efficient algorithm, namely earliest due date (EDD) algorithm. The problem is reduced into single-machine scheduling with due date \(t_{ik}\). The optimal sequences are found by ordering the jobs following the EDD rule. This heuristic yielded better solutions and considered to be computationally faster by 200 times than the heuristics of Miltenburg.

Balinski and Shahidi [5] proposed another type of deviation for the products \(i\) and \(j\), which aims to minimize the variation of production rates from product to products. The production rates of the products \(i\) and \(j\) are measured as \(\frac{x_{ik}}{r_i}\) and \(\frac{x_{jk}}{r_j}\) respectively. If \(\frac{x_{ik}}{r_i} = \frac{x_{jk}}{r_j}\) for all \(i, j\), then the perfection will be gained; however it is very rare in practice. To this end, they proposed the following objective there in [5] to be minimized:

\[
\min \max_{i, j} \frac{x_{ik}}{r_i} - \frac{x_{jk}}{r_j} \quad (3.14)
\]

We have proposed equitably efficient frontier for this objective relating with DAP in Chapter 5 defining the objective function for state to state variation in the number of representatives.

The use of weights \(w_i, i = 1, 2, \ldots, n\) to evaluate the variations of ideal and actual productions plays an important role in balancing the sequence for the model \(i = 1, 2, \ldots, n\). The weighted PRVP objective functions under the constraints (3.3) to (3.6) are defined as follows [26, 105]:

\[
\min \max_{i, k} w_i |x_{ik} - kr_i|, \quad \min \max_{i, k} w_i (x_{ik} - kr_i)^2
\]

and

\[
\min \sum_{k=1}^{D} \sum_{i=1}^{n} w_i |(x_{ik} - kr_i)|, \quad \min \sum_{k=1}^{D} \sum_{i=1}^{n} w_i (x_{ik} - kr_i)^2.
\]

In a recent study [60], both the PRVP objectives are slightly modified with general index as follows: For positive integer \(m\), the bottleneck PRVP and the total PRVP respectively have the forms \(F_m = \min \max_{i, k} |x_{ik} - kr_i|^m\) and \(\sum_{k=1}^{D} \sum_{i=1}^{n} |x_{ik} - kr_i|^m\).
3.1.3 Simultaneous Optimization

The objective functions of the sequencing problem must describe the fact that keeps the effective production as close as possible to the ideal production, and therefore minimize the distance between a feasible sequence and the ideal production. There is no consensus on which distance is the most adequate and many objective functions have been studied in the literature [79]. The detail comparative and computational analysis of the JIT sequencing objectives is presented with four conjectures as open questions in [66].

It is observed that the problems $F_{sum}^s$ and $F_{sum}^e$ have simultaneous features in their solutions directed to optimality. Assuming $B$ as upper bound for the discrepancy values of the objective functions, the main issue is to find minimum $B$ as far as possible. If the deviations in the aforementioned objective functions do not exceed $B$, then the problems are called $B$–bounded problems. The set of all possible solutions of 1–bounded problems is considerably smaller than the one of its unbounded version.

Remaining in the domain of 1–bounded problems, computational improvements can be achieved on exact min-sum methods such as computing the smallest perfect matching which can be done in $O(nD^2 log D)$ for a 1–bounded min-sum problem instead of $O(D^3 log D)$ for its unbounded version [104]. For all instance, it is possible to find a sequence with maximum deviation less than 1. It is interesting to know whether such a 1–bounded sequence can always be found that optimizes total deviation. If there is an optimal solution of $F_{sum}^{s'}$ such that the maximum deviation is less than 1, then this solution is also optimal for $F_{sum}^{s'}$.

The simultaneous optimization of the JIT sequencing problems appears possible when maximum deviation is restricted to be less than 1. In most of the cases with any bound $B$, the simultaneous optimization of the JIT sequencing objectives is hardly reached. However, some motivating conjectures are
posed for simultaneous solutions of the problems.

For any instance of the JIT sequencing problem, we have the conjecture that inspires to the simultaneous optimization [76, 79]: There is a solution optimizing simultaneously the maximum deviation problem $F_{max}^α$ and the sum deviation problem $F_{sum}^{α'}$. The existence of a solution with maximum deviation less than 1 for any instance of the problem [103] indicates that the following conjecture is weaker [104]: There is a sequence that is optimal for $F_{sum}^{α'}$ with maximum deviation less than 1. Another open question that is tested for instances with $D \leq 100$ yielding no example [79] is: For $n = 3$, there are instances such that the problem $F_{sum}^{α'}$ has no optimal solution such that the maximum deviation is lower than 1.

For any instance of the JIT sequencing problem, the final conjecture discussed in [79] is as follows: There is a solution $S^*$ satisfying at least one of the conditions (a) $S^*$ is optimum for both $F_{sum}^{s'}$ and $F_{sum}^{α'}$ (b) $S^*$ is optimum for $F_{sum}^{s'}$ and has maximum deviation less than 1. Thus, the different objectives can be optimized simultaneously with some restriction in the bound. The general cases of the problem are still open. The bicriterion sequencing that considers bottleneck PRVP (i.e., min-max objectives) and total PRVP (i.e., min-sum objectives) simultaneously is discussed in Section 4.6.

### 3.2 Output Rate Variation Problem

#### 3.2.1 Introduction

The mixed-model multi-level JITSP is the output rate variation problem (ORVP). The ORVP consists of several levels in the production supply chain, for example, raw materials → components → subassemblies → products → distribution centers → retailers → customers. In this supply chain system, the multiple copies of different models are produced at the final assembly
For any instance of the JIT sequencing problem, we have the conjecture that inspires to the simultaneous optimization \[76, 79\]: There is a solution optimizing simultaneously the maximum deviation problem \( F_{\text{max}} \) and the sum deviation problem \( F_{\text{sum}}' \). The existence of a solution with maximum deviation less than 1 for any instance of the problem \[103\] indicates that the following conjecture is weaker \[104\]: There is a sequence that is optimal for \( F_{\text{sum}}' \) with maximum deviation less than 1. Another open question that is tested for instances with \( D \leq 100 \) yielding no example \[79\] is: For \( n = 3 \), there are instances such that the problem \( F_{\text{sum}}' \) has no optimal solution such that the maximum deviation is lower than 1. For any instance of the JIT sequencing problem, the final conjecture discussed in \[79\] is as follows: There is a solution \( S^* \) satisfying at least one of the conditions (a) \( S^* \) is optimum for both \( F_{\text{sum}}' \) and \( F_{\text{sum}} \) (b) \( S^* \) is optimum for \( F_{\text{sum}}' \) and has maximum deviation less than 1. Thus, the different objectives can be optimized simultaneously with some restriction in the bound. The general cases of the problem are still open. The bicriterion sequencing that considers bottleneck PRVP (i.e., min-max objectives) and total PRVP (i.e., min-sum objectives) simultaneously is discussed in Section 4.6.

### 3.2 Output Rate Variation Problem

#### 3.2.1 Introduction

The mixed-model multi-level JITSP is the output rate variation problem (ORVP). The ORVP consists of several levels in the production supply chain, for example, raw materials → components → subassemblies → products → distribution centers → retailers → customers. In this supply chain system, the multiple copies of different models are produced at the final assembly level, which is interlinked with several upstream production levels where raw materials are procured, stored and fabricated to produce the final products \[33\] and with several downstream distribution levels where final products are stored and distributed to the retailers and then to the customers.

![Figure 3.2: Multi-level JITPS with Extended Supply Chain Network](image)

The whole body of supply chain consists of inbound logistics along the production levels and outbound logistics along the distribution levels. The Figure 3.2 best illustrates this situation as synchronized view of seven levels of
production and supply chain network. There may be several sublevels in between any two production levels. Therefore, the formulation of the ORVP contains \( L, \ (l = 1, 2, \ldots, L) \) levels.

In the following, we formulate the problem mathematically and explain NP-hard results. Some of the heuristics and sequencing approaches for ORVP are discussed in Chapter 4.

### 3.2.2 Mathematical Formulations

Assume that the mixed-model multi-level JITSP (\( i.e., \) ORVP) consists of \( L \) levels of manufacturing operations, indexed by \( l, \ l = 1, 2, \ldots, L \) with the first product level 1. The number of different part types and the demand of item \( i \) in level \( l \) are denoted by \( n_l \) and \( d_{il} \) respectively, where \( i = 1, 2, \ldots, n_l \). The number of total units of item \( i \) at level \( l \) required to produce one unit of product \( p \) is denoted by \( t_{ilp} \) such that \( d_{il} = \sum_{p=1}^{n_1} t_{ilp}d_{p1} \) is the dependent demand for item \( i \) at level \( l \) determined by the final product demands \( d_{p1}, p = 1, 2, \ldots, n_1 \) and \( l = 1, 2, \ldots, L \). Note that \( t_{ilp} = 1 \) if \( i = p \) and 0 if otherwise.

Finally, \( D_l = \sum_{i=1}^{n_1} d_{il} \) denotes the total demand at level \( l \), and the ratio \( r_{il} = \frac{d_{il}}{D_l} \) gives the demand rate for item \( i \) of level \( l \) such that \( \sum_{i=1}^{n_l} r_{il} = 1 \) at each level \( l = 1, 2, \ldots, L \). It is noteworthy that the model of ORVP is assumed to be non-preemptive; that is, once commenced production of a product at level 1 must be completed prior to switch into another unit. This creates the concept of various stages or cycles in the production system. The production schedule at level 1 consists of \( D_1 \) stages in total and at each stage a single unit of an end-product can be processed. An item is said to be in stage \( k \), \( (k = 1, 2, \ldots, D_1) \), if \( k \) units of product have been produced at level 1 and there will be \( k \) complete units of various products \( p \) at level 1 during the first
Let $x_{ilk}$ be the necessary quantity of item $i$ produced at level $l$ during the time units $1$ through $k$ and $y_{lk} = \sum_{i=1}^{n_i} x_{ilk}$ be the cumulative quantity of item $i$ produced at level $l$ during the same time units such that $y_{lk} = \sum_{i=1}^{n_i} x_{ilk} = k$.

Due to the pull nature of the JITPS, the particular combination of the highest level products produced during the $k$ time units (the $x_{p1k}$ values) determines the necessary cumulative production at every other level. Thus, the required cumulative production for item $i$ at level $l$ with $l \geq 2$ through $k$ time units is given by $x_{ilk} = \sum_{p=1}^{n_i} t_{ilp} x_{p1k}$. For a unimodal convex penalty function $F_i$, $i = 1, \ldots, n_i$ with minimum 0 at 0, the maximum deviation and the sum deviation multi-level JIT sequencing problems in mixed-model systems (i.e., ORVP) are mathematically formulated to minimize the objectives $Z_{\text{max}}$ and $Z_{\text{sum}}$ as the followings [31, 71, 86]:

$$Z_{\text{max}} = \min_{i,l,k} \max_{p} F_i(x_{ilk} - y_{lk} r_{il})$$

$$Z_{\text{sum}} = \min \sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_i} F_i(x_{ilk} - y_{lk} r_{il})$$

subject to

$$x_{ilk} = \sum_{p=1}^{n_i} t_{ilp} x_{p1k}, \quad i = 1, \ldots, n_i; \quad l = 1, \ldots, L; \quad k = 1, \ldots, D_1 \quad (3.17)$$

$$y_{lk} = \sum_{i=1}^{n_i} x_{ilk}, \quad l = 2, 3, \ldots, L; \quad k = 1, \ldots, D_1 \quad (3.18)$$

$$x_{p1k} \geq x_{p1(k-1)}, \quad p = 1, 2, \ldots, n_1; \quad k = 1, 2, \ldots, D_1 \quad (3.20)$$

$$x_{p1D_1} = d_{p1}, \quad x_{p10} = 0, \quad p = 1, 2, \ldots, n_1 \quad (3.21)$$

$$x_{ilk} \geq 0 \text{ integer}, \quad i = 1, \ldots, n_i; \quad l = 1, \ldots, L; \quad k = 1, \ldots, D_1 \quad (3.22)$$

$k$ time units.
Here the constraint (3.17) ensures that the necessary cumulative production of part \( i \) of level \( l \) by the end of time unit \( k \) is determined explicitly by the quantity of products produced at level 1. Constraints (3.18) and (3.19) show the total cumulative production of level \( l \) and level 1 respectively during the time slots 1 through \( k \). Constraint (3.20) ensures that the total production of every product over \( k \) time units is a non-decreasing function of \( k \). Constraint (3.21) guarantees that the demands for each product are met exactly, and (3.22) is the integrality constraint. The constraints (3.19), (3.20), (3.22) jointly ensure that exactly one unit of a product is scheduled during one time unit in the product level.

The particular cases of the objectives (3.15) and (3.16) are studied in literature [31, 33] as absolute and squared deviation objectives in both cases as follows:

\[
Z_{max}^a = \min \max_{i,l,k} |x_{ilk} - y_{ilk}r_{il}| \quad (3.23)
\]

\[
Z_{max}^s = \min \max_{i,l,k} (x_{ilk} - y_{ilk}r_{il})^2 \quad (3.24)
\]

\[
Z_{min}^a = \min \sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_l} |x_{ilk} - y_{ilk}r_{il}| \quad (3.25)
\]

\[
Z_{min}^s = \min \sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_l} (x_{ilk} - y_{ilk}r_{il})^2 \quad (3.26)
\]

The proper use of appropriate weights in the sequencing objectives is more fruitful feature. These weights show the relative importance of the model that affects the sequencing of the model into which that part is to be assembled [105]. The weights can be used to smooth the variability and to prevent lower level parts to be dominant over higher level parts in the measures at different levels. The selection of weights is based on the total production at various levels, the relative importance of having good schedules at the various levels and the numerical values assigned to the weights [75, 86].

The weight given to model \( i \) at level \( l \) is denoted by \( w_{il} \). With this notation,
the above objective functions in particular cases can be reformulated in the weighted case of ORVP as below [33, 60]:

\[
\tilde{Z}_{\text{max}}^a = \min_{i,l,k} \max w_{il} |x_{ilk} - y_{ilk}r_{il}| \quad (3.27)
\]

\[
\tilde{Z}_{\text{max}}^s = \min_{i,l,k} \max w_{il}(x_{ilk} - y_{ilk}r_{il})^2 \quad (3.28)
\]

\[
\tilde{Z}_{\text{min}}^a = \min \sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_l} w_{il} |x_{ilk} - y_{ilk}r_{il}| \quad (3.29)
\]

\[
\tilde{Z}_{\text{min}}^s = \min \sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_l} w_{il}(x_{ilk} - y_{ilk}r_{il})^2 \quad (3.30)
\]

The ORVP is a nonlinear integer programming problem, whose objective functions capture the sequence dependent nature of the schedule for lower parts. The required cumulative productions \(x_{ilk}\)'s, \(l > 1\) are calculated directly from the assembly sequence of the products \(x_{i1k}\)'s, and the desired production goal for model \(i\) in level \(l\) is calculated as the ideal proportion \((r_{il})\) of the total cumulative production quantity \(y_{ilk}\) of level \(l\). Balanced schedules are generated by keeping the required production of all parts and products as close to this goal as possible.

The min-max objectives of ORVP aim to find a smooth schedule in every time period for every output. This is the basic concept underlying Toyota’s sequencing algorithm [89]. Moreover, the value of the objective function \(Z_{\text{max}}^a\) represents an applicable physical application, providing the maximum overproduction or underproduction (the maximum inventory or shortage) from the desired quantity of production that occurs at any time in the schedule. This information may be used to determine the number of kanbans (or the necessary safety stocks) used [75]. The min-sum objectives of ORVP seek optimal schedules that may have relatively large deviation in a single period or for a certain output while having the lowest possible total deviation.

35
3.2.3 NP-Hardness

For an input size \( n \) of a problem \( P \), a generally accepted minimum requirement for an algorithm to be considered efficient is that its running time is polynomial in \( n \), denoted by \( O(n^c) \) for some constant \( c \). Scientific researchers have recognized that not all problems can be solved this quickly. Due to this scientific contribution, we are able now to figure out exactly which problems could be solved and which ones couldn’t be solved. There are several so-called NP-hard problems, which most people believe cannot be solved in polynomial time, even though nobody can prove a super-polynomial lower bound. Circuit satisfiability is a good example of a problem that we don’t know how to solve in polynomial time.

A decision problem is a problem whose output is a single boolean value: yes or no, true or false, on or off etc. Based on this definition, there are three classes of decision problems: \( P \) (solvable in polynomial time), \( NP \) (Non-polynomial) and Co-\( NP \) (complements of problems in \( NP \)). The notation \( NP \) stands for \textit{nondeterministic polynomial time}, since originally \( NP \) was defined in terms of nondeterministic machines. Regarding this issue, whether \( P \) is equal to \( NP \) is one of the stunning conjecture. For a detail literature of computational complexity classes, we recommend [45, 46, 80, 96, 110].

The crux of a combinatorial problem is to develop an algorithm that guarantees identifying an optimal solution for every instance of the problem. Unfortunately as illustrated above, not all combinatorial problems possess an algorithm with small amount of computer time. For example, Steiner tree problem, 3-partition problem, Exact 3-dimensional matching problem are some intractable problems.

The ORVP with the sum of the square deviation objective has been shown to be NP-hard in the ordinary sense [76]. This result has been achieved by reducing the scheduling around the shortest job (SASJ) problem to the
3.2.3 NP-Hardness

For an input size \( n \) of a problem \( P \), a generally accepted minimum requirement for an algorithm to be considered efficient is that its running time is polynomial in \( n \), denoted by \( O(n^c) \) for some constant \( c \). Scientific researchers have recognized that not all problems can be solved this quickly. Due to this scientific contribution, we are able now to figure out exactly which problems could be solved and which ones couldn’t be solved. There are several so-called NP-hard problems, which most people believe cannot be solved in polynomial time, even though nobody can prove a super-polynomial lower bound. Circuit satisfiability is a good example of a problem that we don’t know how to solve in polynomial time.

A decision problem is a problem whose output is a single boolean value: yes or no, true or false, on or off etc. Based on this definition, there are three classes of decision problems: \( P \) (solvable in polynomial time), \( NP \) (Non-polynomial) and Co-\( NP \) (complements of problems in \( NP \)). The notation \( NP \) stands for nondeterministic polynomial time, since originally \( NP \) was defined in terms of nondeterministic machines. Regarding this issue, whether \( P \) is equal to \( NP \) is one of the stunning conjecture. For a detail literature of computational complexity classes, we recommend [45, 46, 80, 96, 110].

The crux of a combinatorial problem is to develop an algorithm that guarantees identifying an optimal solution for every instance of the problem. Unfortunately as illustrated above, not all combinatorial problems possess an algorithm with small amount of computer time. For example, Steiner tree problem, 3-partition problem, Exact 3-dimensional matching problem are some intractable problems.

The ORVP with the sum of the square deviation objective has been shown to be NP-hard in the ordinary sense [76]. This result has been achieved by reducing the scheduling around the shortest job (SASJ) problem to the 36 ORVP. The scheduling around the shortest job problem finds a schedule, on a single machine, of independent jobs \( i, i = 1, 2, \ldots, n \), that minimizes the sum \( \sum_{i=1}^{n} (C_i - C_1)^2 \), where \( C_i \) is completion time of job \( i, i = 1, 2, \ldots, n \) with processing times \( p_1 \leq p_2 \leq \ldots \leq p_n \).

The SASJ problem is NP-hard in the ordinary sense [72]. Moreover, the min-sum ORVP problem is computationally more difficult and the results established so far on the completion time variance minimization problem indicate that even special cases of ORVP are NP-hard.

Furthermore, bottleneck ORVP with absolute-deviation objective that considers only two levels of production has been proved to be NP-hard in the strong sense. An instance of the 3-partition problem can be reduced into an instance of ORVP with two levels in pseudo-polynomial time [60, 75]. The 3-partition problem is to decide whether a given multiset of integers can be partitioned into triples that all have the same sum. That is, for \( 3m \) integers, is there a partition \( \{A_1, A_2, \ldots, A_m\} \) of the set \( \{1, 2, \ldots, 3m\} \) such that \( \sum_{i \in A_i} a_i = B, 1 \leq i \leq m \), where \( a_i \) is a positive integer, \( 1 \leq i \leq 3m \) and \( B \) is a bound such that \( \sum_{i=1}^{3m} = mB, \frac{B}{4} < a_i < \frac{B}{2} \)? The well-known fact is that the 3-partition problem is strongly NP-complete [60, 96].
Chapter 4

Just-in-Time Sequencing Approaches

Chapter 4 is based on the papers A, C, G, L, N, S.

It appears in the literature that the problem of optimally sequencing mixed-model assembly systems in JIT environment has been the subject of extensive research for more than five decades. A number of exact and heuristic solution approaches have been developed for several well-known sequencing approaches like mixed-model sequencing \[108, 119\], car sequencing \[97, 102\] and level scheduling \[76, 91\]. An intensive overview on model sequencing is presented by Boysen et al. \[19, 20\]. However, these traditional sequencing approaches are not sufficient to cover the recent industrial trends and their dynamic complexity.

Heuristics are common-sense rules or approximate techniques which intend to increase the probability of solving optimization problems. Moreover, the term heuristic is used for algorithms which find solutions among all possible ones, but do not guarantee about the best one; therefore heuristics are considered as approximate and not accurate algorithms. This means that the heuristic approaches provide comparatively good solutions, not necessarily optimal.
Chapter 4

Just-in-Time Sequencing Approaches

Chapter 4 is based on the papers A, C, G, L, N, S.

It appears in the literature that the problem of optimally sequencing mixed-model assembly systems in JIT environment has been the subject of extensive research for more than five decades. A number of exact and heuristic solution approaches have been developed for several well-known sequencing approaches like mixed-model sequencing [108, 119], car sequencing [97, 102] and level scheduling [76, 91]. An intensive overview on model sequencing is presented by Boysen et al. [19, 20]. However, these traditional sequencing approaches are not sufficient to cover the recent industrial trends and their dynamic complexity.

Heuristics are common-sense rules or approximate techniques which intend to increase the probability of solving optimization problems. Moreover, the term heuristic is used for algorithms which find solutions among all possible ones, but do not guarantee about the best one; therefore heuristics are considered as approximate and not accurate algorithms. This means that the heuristic approaches provide comparatively good solutions, not necessarily optimal.
Usually heuristic algorithms are used for problems that cannot be easily solved. A good heuristic is a simple heuristic with good average performance and with reasonable time complexity. Some of the mixed-model sequencing heuristics are developed by academic research scientists, while others are emerging as a result of practical applications.

In this chapter, we present some of the solution strategies for the JITSPs. Heuristic approaches for PRVP and ORVP are explained in Section 4.1 and Section 4.2 respectively providing dynamic programming solutions for each problem. Perfect matching approach for bottleneck PRVP and assignment solution for total PRVP are discussed and analysed in Section 4.3 and Section 4.4 respectively. Section 4.5 briefly describes the bottleneck PRVP with general index followed by bicriteria sequencing in the final Section of this Chapter.

4.1 Heuristics for Product Rate Variation

There are several sequencing heuristics that give near-optimal solutions for minimizing the single-level variation at final stage of production lines [118]. We present some single-level sequencing heuristics and their modified versions in the following Subsections.

4.1.1 Miltenburg’s Heuristics

Miltenburg proposed three algorithms and two heuristics with their mutual assimilation to solve the PRVP [85]. The Miltenburg’s heuristic procedures starts with algorithm 1 (MA1) to find the nearest integer point \(x_{1k}, x_{2k}, \ldots, x_{nk}\) at each stage \(k\) to \(kr_1, kr_2, \ldots, kr_n\). MA1 considers each time unit of the sequence independently. For each time unit, a cumulative production \(x_{ik}\) of model \(i, i = 1, 2, \ldots, n\) is obtained. If production re-
Usually heuristic algorithms are used for problems that cannot be easily solved. A good heuristic is a simple heuristic with good average performance and with reasonable time complexity. Some of the mixed-model sequencing heuristics are developed by academic research scientists, while others are emerging as a result of practical applications.

In this chapter, we present some of the solution strategies for the JITSPs. Heuristic approaches for PRVP and ORVP are explained in Section 4.1 and Section 4.2 respectively providing dynamic programming solutions for each problem. Perfect matching approach for bottleneck PRVP and assignment solution for total PRVP are discussed and analysed in Section 4.3 and Section 4.4 respectively. Section 4.5 briefly describes the bottleneck PRVP with general index followed by bicriteria sequencing in the final Section of this Chapter.

4.1 Heuristics for Product Rate Variation

There are several sequencing heuristics that give near-optimal solutions for minimizing the single-level variation at final stage of production lines [118]. We present some single-level sequencing heuristics and their modified versions in the following Subsections.

4.1.1 Miltenburg’s Heuristics

Miltenburg proposed three algorithms and two heuristics with their mutual assimilation to solve the PRVP [85]. The Miltenburg’s heuristic procedures starts with algorithm 1 (MA1) to find the nearest integer point $x_{1k}, x_{2k}, \ldots, x_{nk}$ at each stage $k_1, k_2, \ldots, k_r$. MA1 considers each time unit of the sequence independently. For each time unit, a cumulative production $x_{ik}$ of model $i$, $i = 1, 2, \ldots, n$; $k = 1, 2, \ldots, D$ and if too small, $x_{ik}$ is incremented with the smallest $x_{ik} - kr_i$. The MA1 is fast enough to solve the problem with a realistic size only, so that other algorithms are also proposed in [85].

The second algorithm with the first heuristic solves squared deviation problem by testing the feasibility of the schedule. The third algorithm determines whether the schedule is feasible. The first heuristic is used with third algorithm (MA3H1) to calculate an entire schedule for the mixed-model JIT production system considering the product rates, not the parts usage rates. MA3H1 begins with MA1 to obtain a trial sequence which is revised to obtain feasibility. It is one-stage myopic heuristic doing one calculation for each product and then making a selection with complexity $O(nD)$ for each stage. But it does not consider the effect of its current decision on the variation in future stages.

Due to the myopic nature of the first heuristic, Miltenburg [85] further developed the improved two-stage second heuristic with the complexity $O(n^2D)$ for each stage, which together with the third algorithm (MA3H2) approximates the variability over the two stages and schedules in such a way that this variability is as small as possible. Each stage is rescheduled using the same criteria as in MA3H1. The only difference is that each model is selected so that the variation in the product usage rates over the stages $k$ and $k + 1$, $k = 1, 2, \ldots, D - 1$ is a minimum.

MA3H2 has better performance than MA3H1 due to the reduction of myopia in MA3H2. The myopia lies in the fact that MA3H1 only takes one step. Taking two steps into account, the myopia can be reduced [14]. The analysis of the heuristics in [36, 107] showed that MA3H2 is of the highest quality heuristic. A simpler heuristic as good as MA3H2 in solution quality for the problem with squared or absolute deviation objectives and better in computational time has been established in the literature [36, 37, 60].

41
Due to the large size of the products and their units $D$, the impression of Miltenburg’s heuristics is not so effective. The various heuristics utilizing large-size problems and representing realistic situations are compared in [40] and examined relative performance of those mixed-model sequencing heuristics based on their ability to develop a sequence for final assembly which smoothes out the rate of use of each component part feeding the assembly line.

The MA3H2 has been modified using appropriate weights as useful alternatives for frequent updates of sequencing [26]. In this modified heuristic, modification occurs in MA1 to find a trial sequence. Modification is that the coordinate $x_{ik}$ is found with the smallest $w_i(x_{ik} - kr_i + \frac{1}{2})$, $i = 1, 2, \ldots, n; k = 1, 2, \ldots, D$ to get incremented and with the largest $w_i(x_{ik} - kr_i - \frac{1}{2})$ to get decremented. The term $w_i$ is the weight issued to model $i$, $i = 1, 2, \ldots, n$, [26]. Another modification occurs in second heuristic in [85]. In this modification, model $i$ is scheduled for stage $k + 1$ with the lowest $w_i(x_{ik} - (k + 1)r_i + \frac{1}{2})$, $i = 1, 2, \ldots, n; k = 1, 2, \ldots, D$.

### 4.1.2 Earliest Due Date Rule

Earliest due date (EDD) rule is a two-stage heuristic [56] similar to MA3H2. In the EDD heuristic, an ideal time $\frac{2^{j-1}}{2r_i}$, $i = 1, 2, \ldots, n; k = 1, 2, \ldots, D$ is defined. The ideal time means the time at which the $j^{th}$ copy of model $i$ is produced known as its due date. The objective of the EDD rule is to minimize the deviation between the time at which a unit of a model is actually produced and the time at which the unit of the model is needed to be produced. The objective is mathematically different but intuitively similar to the objective established in [86].

Each copy of the models is sequenced using the EDD rule, which is a special case of the earliness and the tardiness sequencing problem with a common
processing time \[44\]. The time complexity of this rule is \(O(nD)\). It has been established that MA3H2, Ding & Cheng’s heuristic \[36, 37\] and EDD rule obtain good quality solution \[40\]. The literature compares several heuristics utilizing realistic and large-size problems. The EDD rule is modified by giving weights in the objective function.

The **Modified EDD rule** is the modification of EDD rule with weighted objective of the deviation between the time at which a unit of a model is actually produced and the time at which the unit of the model is needed to be produced. A copy \(j\) of model \(i\) is sequenced at stage \(k, k = 1, 2, \ldots, D\) with the lowest \(w_i(\frac{2j-1}{2r_i} - k - \frac{1}{2}), i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i\) \[26\].

### 4.1.3 Two-stage Algorithm

In the two-stage heuristic algorithm, a trial sequence is revised to obtain a feasible sequence. The objective function is to minimize the following sum:

\[
\min \sum_{k=1}^{D} \sum_{i=1}^{n} (x_{ik} - k r_i)^2 \tag{4.1}
\]

A model at the stage \(k\) is selected in such a way that the procedure will minimize the objective function at the next two stages \(k\) and \(k+1\). The heuristic determines a model \(i\) with the lowest \(x_{i(k-1)} - (k + \frac{1}{2}) r_i\). The model \(\hat{i}, i \neq \hat{i}\) is rescheduled at stage \(k\) if \(x_{i(k-1)} - (k + \frac{1}{2}) r_i - x_{\hat{i}(k-1)} + (k + \frac{1}{2}) r_\hat{i} > \frac{1}{2} (r_i - r_\hat{i})\) \[36, 37\].

The heuristic is as good as MA3H2 and is faster in computation with complexity \(O(nD)\). This is easy to be coded into a computer program and requires less memory space \[36\]. Note that the two-stage algorithm can be modified with \(x_{i(k-1)} - (k + a) r_i, a \in [0, 1]\) instead of \(x_{i(k-1)} - (k + \frac{1}{2}) r_i\) applying the weight factor. The modified two-stage algorithm outperforms the two-stage algorithm \[63\].
In the **modified two-stage algorithm**, the modification occurs in the selection of model \(i\) at stage \(k\) with the lowest \(w_i(x_i(k-1)) - (k + \frac{1}{2})r_i + \frac{1}{2}\), \(i = 1, 2, \ldots, n\); \(k = 1, 2, \ldots, D\) and model \(\hat{i}\), \(i \neq \hat{i}\) is rescheduled at stage \(k\) if \(w_i(x_i(k-1)) - (k + \frac{1}{2})r_i - w_i(x_i(k-1) + (k + \frac{1}{2})r_i) > \frac{1}{2}(w_i r_i - w_i r_i)\), \(i = 1, 2, \ldots, n\); \(k = 1, 2, \ldots, D\) otherwise schedule model \(i\) at stage \(k\) [26].

### 4.1.4 Final Assembly Sequencing Algorithm

A workable sequence in final assembly line should have an reasonable level of intermixing of products together with an acceptable number of required set-ups. So sequencing products in final assembly line is an important task to balance the production system. The final assembly sequencing (FAS) is an efficient local search procedure based on selective pairwise interchanges, which solves a lexicographic minmax formulation of FAS problem [53]. The FAS algorithm is a local search heuristic as a solution method that attempts to swap the order of assembly of a pair of models. It adopts swapping to swap the order of assembly of a pair models. The swapping is repeatedly applied to an infeasible sequence until no further swapping is required [53].

In this FAS procedure, the deviations are sorted in non-increasing order that the algorithm minimizes the first element, then the second element is treated as the first element. The algorithm consists of three modules: a pre-processing module, a feasibility module and an optimization module [53]. The pre-processing module reduces the size of the problem. The feasibility module finds an initial feasible sequence and the optimization module naturally works for optimality.

The FAS algorithm provides near-optimal sequence for realistic-size problems in a reasonable running time. It may be extended considering release date and due date constraints. Note that if a tie occurs while implementing any of the above algorithms, then the tie is broken arbitrarily.
Dynamic programming (DP) procedure deals with optimal decision making problems in which events can be separated into sequential stages. DP was developed by R Bellman for discrete types of decisions [17]; however it can be applied for the problems of continuous nature as well. The use of heuristics and bounded dynamic programming to solve the Monden problem can be found in [14].

A dynamic programming procedure to determine the optimal JIT production schedule for single-level min-sum problem is presented in [87], considering usage and loading goals simultaneously. The usage goal maintains a constant rate of usages of all items in the system. The loading goal does smooth the work load on the final assembly process to reduce the chance of the production delays and stoppages by balancing the products having long production times with the products having relatively short production times. The former goal is considered more important than the latter one for most of the manufacturers [89].

First we formalize the DP formulation of PRVP: Taking the notations as in Section 3.1, we let \( t_i \) be the time required to produce one unit of product \( i \) and define usage and loading variabilities at stage \( k \). If a schedule for the first \( k \) stages (i.e., \( x_{ik}, i = 1, 2, \ldots, n \)) is known, then the usage variability at stage \( k \) is defined by \( U_k = \sum_{i=1}^{n} (x_{ik} - kr_i)^2 \). To satisfy this goal, the objective is to sequence the assembly process so that the proportion of product \( i \) produced relative to the total production up to stage \( k \) is as close to \( r_i \) as possible. If a schedule for the first \( k \) stages (i.e., \( x_{ik}, i = 1, 2, \ldots, n \)) is given, then the loading goal is defined by \( L_k = \sum_{i=1}^{n} t_i^2 (x_{ik} - kr_i)^2 \).

The selection of integer \( x_{ik} \) to minimize \( L_k \) prevents the products with relatively longer production times from being scheduled consecutively. More-
over, the products with longer production times are quickly balanced off with products having relatively short production times.

In this case, the joint problem is to seek the integer \( x_{ik}, i = 1, 2, \ldots, n; k = 1, 2, \ldots, D \) that minimizes the following weighted sequencing objective

\[
\text{minimize } \sum_{k=1}^{D} (\alpha_U U_k + \alpha_L L_k) \quad (4.2)
\]

under the constraints (3.3) to (3.6), where \( \alpha_U \) and \( \alpha_L \) are the relative weights for the usage and loading goals respectively. If \( \alpha_U = 1 \) and \( \alpha_L = 0 \), then the problem is converted to the usage problem. If \( \alpha_U = 0 \) and \( \alpha_L = 1 \), then the problem is converted to the loading problem. The objective function (4.2) can be reduced to an equivalent loading problem by taking \( v_k \) as joint variability at stage \( k \).

That is, we have

\[
v_k = \sum_{k=1}^{D} (\alpha_U U_k + \alpha_L L_k)
\]

\[
= \alpha_U \sum_{i=1}^{n} (x_{ik} - kr_i)^2 + \alpha_L \sum_{i=1}^{n} t_i^2 (x_{ik} - kr_i)^2
\]

\[
= \sum_{i=1}^{n} (\alpha_U + \alpha_L t_i^2) (x_{ik} - kr_i)^2
\]

\[
= \sum_{i=1}^{n} T_i^2 (x_{ik} - kr_i)^2
\]

where \( T_i^2 = \alpha_U + \alpha_L t_i^2 \).

and \( T_i \) is referred as the implied production time for product \( i \). Hence the modified total PRVP objective equivalent to (4.2) is as the following one.

\[
\text{minimize } \sum_{k=1}^{D} \sum_{i=1}^{n} T_i^2 (x_{ik} - kr_i)^2 \quad (4.3)
\]

Now we discuss the DP procedure for the joint problem: The DP procedure
enables to obtain optimal JIT schedule for practical-sized problems [87]. Let \( d=(d_1,d_2,\ldots,d_n) \) be the demand vector and \( X=(x_1,x_2,\ldots,x_n) \) be the product vector in which each \( x_i \) is a non-negative integer representing the production of exactly \( x_i \) units of product \( i \), \( x_i \leq d_i, \forall i \). Let \( e_i \) be unit vector with \( n \) entries, all of which are zero except a single one in \( i^{th} \) position.

Products in \( X \) can be produced in first \( k \) periods if \( k=|X|=\sum_{i=1}^{n} x_i \).

Let \( f(X) \) be minimal total variation of any schedule where the products in \( X \) are produced during the first \( k \) stages. For copy \( j \), letting \( g(X) = \sum_{j=1}^{n} T_j^2 (x_j - kr_j)^2 \), the following DP recursion process holds for \( f(X) \):

\[
f(X) = f(x_1,x_2,\ldots,x_n) = \min \{ f(X-e_i) + g(X) : i = 1,2,\ldots,n; x_i - 1 \geq 0 \}\]

and \( f(\emptyset) = f(X:x_i = 0, \forall i) = f(0,0,\ldots,0) = 0 \).

It is obvious that \( f(X) \geq 0 \) and hence \( g(X: \forall x_i = d_i) = 0 \)

The following theorem states the time complexity and the space complexity of the above DP procedure [87].

**Theorem 4.1** The DP recursion procedure solves the JIT scheduling problem (4.3) in \( O\left(n \prod_{i=1}^{n} (d_i + 1)\right) \) time and \( O\left(\prod_{i=1}^{n} (d_i + 1)\right) \) space.

Note that total number of feasible schedule is \( \frac{D!}{d_1!d_2!\cdots d_n!} \), which is considerably larger than the number of stages in the DP recursion. Moreover, we have the inequality

\[
\prod_{i=1}^{n} (d_i + 1) \leq \left(\frac{d_1+d_2+\ldots+d_n+n}{n}\right)^n = \left(\frac{D+n}{n}\right)^n.
\]

Thus, the growth rate of the number of sets is polynomial in \( D \) although it is exponential with \( n \). This indicates that the DP procedure is effective.
for small $n$ even with large $D$ and hence it significantly reduces the space complexity by generating the sets in order of increasing cardinality. That is, all sets $X$ with $|X| = k$ are generated before proceeding to the set $X$ with $|X| = k + 1$. The optimization process that run in polynomial time in $D$ for all products, produced over a given time horizon is presented for two instances of the single-level problem [87].

### 4.2 Heuristics for Output Rate Variation

A number of sequencing methods as heuristics has been developed and reported with comparison in the literature due to the popularity of JITPS evolved during the 1980s [40, 107, 108]. A complex heuristic for selecting the production sequence when the objective is to minimize the chance of stopping the line due to overloading individual stations is proposed in [94]. In this heuristic, Okamura and Yamashina [94] suggest a procedure, which uses many different initial sequences. For each initial sequence, an improvement routine is applied in which jobs are moved until no improvement occurs, followed by an interchange of jobs until no improvement occurs. The best of the several sequences is the solution. They present empirical results for problems with up to 100 jobs, which suggest that the heuristic perform almost as well as a branch and bound procedure with a CPU time trap of 2 seconds (see [61] also).

Monden [89] developed the two greedy heuristics at Toyota, which he referred as goal chasing methods: GCM I and GCM II (see [65] too). The heuristics GCM I and GCM II, designed with product level and sub-assembly level, constructed a sequence filling one position at a time from first slot to the last one, considering the variability at the sub-assembly level. In comparison of GCM I, the GCM II represented a decrease in computational time, since the sum is formed only on the components of a given product [107]. However,
the comparative research in [107] and in [108] showed that GCM I performed better than GCM II when compared on the basis of maintaining a constant usage of component parts. These heuristics has been found to yield very good results in the Toyota [62].

Hyundai’s heuristic (HH) used an alternative way, which was developed to approximate the result given by GCM I while reducing the steps of computation. Duplaga and Bragg [40] concluded that the reduction in computational effort related to HH may be significant in situations similar to automobile assembly where many options and choices are available for final product configurations.

The GCM has been advanced to the Extended Goal Chasing Method (EGCM) to consider all levels in a multi-level production system [86] and introduced another polynomial heuristic to reduce the myopic nature of the previous heuristic. Moreover, the myopic nature of the GCM I has been reduced and an exact procedure based on the bounded dynamic programming is developed in [14].

In the following Subsections, we briefly discuss on the goal chasing heuristics and on the other two solution procedures as well, namely the pegging assumption and DP procedure.

### 4.2.1 Goal Chasing Method I

The goal chasing method I (GCM I) was developed and used by Toyota to schedule automobile final assembly lines. It constructs a sequence filling one time unit at a time from first slot to the last one. This method is designed with the two levels: the product level and the sub-assembly level, considering the variability at the sub-assembly level only, whereas the variability is ignored at the final level [86].

For a stage $k$, the objective function used in GCM I to schedule the product
\[ i \text{ is} \]
\[
\text{minimize} \left[ GCM \ I = \sum_{i=1}^{n_2} \left[ x_{i2k} + t_{i2k} - y_{2k}r_{i2} \right]^2 \right] \quad (4.4)
\]

The GCM I is a myopic heuristic. This heuristic yields infeasible sequence frequently but if it yields a feasible sequence, then the sequence is necessarily optimal too [107]. The time complexity of GCM I is \( O(n_1nD) \).

### 4.2.2 Goal Chasing Method II

As in the case of GCM I, the GCM II constructs a sequence filling one time unit at a time from first slot to last one. The GCM II is designed to decrease computational time because the sum is formed only on the components of a given model [86]. This indicates that the computational time can be considerably saved if a model encompasses only a small fraction of the total number of parts [108].

For stage \( k \), the objective function used in GCM II to schedule the product \( i \) is
\[
\text{minimize} \left[ GCM \ II = \sum_{i \in C} \left[ x_{i2k} - y_{2k}r_{i2} \right]^2 \right] \quad (4.5)
\]

where \( C \) is the set of components of a given model. If \( C \) contains a small fraction of total number of components, the computational time is substantially reduced. This heuristic is also myopic and frequently generates infeasible sequence.

The goal chasing method has been extended in [86] to consider all levels, which is called Extended Goal Chasing Method (EGCM). It can be said that the GCM I and GCM II are special cases of the EGCM.
4.2.3 Extended Goal Chasing Method

The extended goal chasing method (EGCM) is also a heuristic for multi-level problem since it includes more levels [86]. For a stage $k$, the objective function used in EGCM to schedule the product $i$ is

$$\text{minimize } \left[ \text{EGCM} = \sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_l} w_l (x_{ilk} - y_{lk}r_{il})^2 \right]$$

(4.6)

where the weight $w_l$ determines the relative importance of a level $l$. The heuristic sequences model $i$ at time unit $k$ with minimum

$$\sum_{l=1}^{L} \sum_{i=1}^{n_l} (x_{ilk} + t_{ilk} - y_{lk}r_{il})^2$$

(4.7)

This is also a myopic polynomial heuristic. There exist two heuristics for the solution of the problem in [86].

4.2.4 The Pegging Assumption

The output rate variation problem is NP-hard combinatorial problem. However, this can be solved in pseudo-polynomial time under the pegging assumption, which separates each part at the lower production levels into distinct groups for each product into which that part will be assembled. The pegging process reduces the ORVP into weighted case of the product rate variation problem [52, 105].

In the pegging assumption, parts of output $i$ at level $l$, $l = 2, 3, \ldots, L$, are dedicated or pegged to be assembled into the particular model at level 1. The parts dedicated to be assembled into the different models are distinct in pegging $i.e., h \neq p$ implies $t_{ilh} \neq t_{ilp}$ for each output $i$ at level $l$. Pegging is useful for high quality model production because high quality parts are required for high quality model and such parts can be used under this assumption [60].
The mathematical formulation of pegging in a JIT production environment has been firstly developed in [52], where some heuristic procedures for the pegged multi-level min-sum model are also presented.

The sequencing model of the pegged ORVP with absolute deviation objective [105, 60] is to minimize the following weighted deviation:

\[
\min \max_{h,i,l,k} \{w_{h1} |x_{h1k} - kr_{h1}|, w_{il} |x_{h1k}t_{ilh} - kt_{ilh}r_{h}|\} \quad (4.8)
\]

where \( h = 1, 2, \ldots, n_{1}; \ i = 1, 2, \ldots, n_{l}; \ k = 1, 2, \ldots, D_{1}; \ l = 2, 3, \ldots, L \)

subject to the constraints (3.17) to (3.22).

For \( l = 1, 2, \ldots, L, t_{ilh} = 1 \) if \( i = h \) and 0 otherwise, the objective function is reduced to \( \min \max_{h,i,l,k} \{w_{il}(t_{ilh}) |x_{i1k} - kr_{i1}|\} \). With \( \tilde{w}_{i}^* = \max_{i,l} \{w_{il}(t_{ilh})\} \), pegged ORVP is transformed into the following formulation:

\[
\min \max_{i,k} \tilde{w}_{i}^* |x_{ik} - kr_{i}| \quad (4.9)
\]

with constraints (3.3) to (3.6).

Clearly, this is the weighted product rate variation problem formulation [105]. The pegged ORVP with total deviation objective can analogously be reduced to a weighted PRVP with total deviation objective [105]. The optimal schedules for the weighted PRVP with total deviation objective can be obtained using the assignment approach [76, 71, 74].

### 4.2.5 Dynamic Programming Solution

The efficient algorithms for the solutions of ORVP are unlikely to exist due to the NP-hardness of the problem. Nevertheless, the dynamic programming (DP) procedure gives rise to optimal solutions [75] for small number of products. The DP algorithm has been applied for the problem with the objective that simultaneously minimizes the variability in the usage of parts and
smooths the workload in the final assembly process [87], which has been briefly discussed in Subsection 4.1.5 for PRVP.

The DP procedure for ORVP is developed in [75], which is polynomial in $D_1$ and consequently, seems to be effective for small number of products $n_1$ even when the total product demand $D_1$ is large. During the enumeration process, an excessive amount of time or space is reduced by using some fast heuristic as a filter which eliminates any states from DP’s state space that would lead to no optimality. Two myopic heuristics to generate the filter are proposed in [75]. If the heuristics yield near-optimal sequences, then the state space size can be reduced.

The weighted case of output rate variation problem with the two sequencing objective functions $\max_{i,l,k} w_{il} |x_{ilk} - y_{ilk} r_{il}|$ and $\sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_1} w_{il} (x_{ilk} - y_{ilk} r_{il})^2$ can be concisely transformed into the matrix representation and can be implemented the transformation for the solution of ORVP using DP procedure [75].

First we consider the min-max objective function $\max_{i,l,k} w_{il} |x_{ilk} - y_{ilk} r_{il}|$ and denote the deviation matrix $\Gamma = [\gamma_{il}]_{n \times n_1}$ with $n = \sum_{l=1}^{L} n_l$, where $\gamma_{ilp}$ represents the $(\sum_{m=1}^{l-1} n_m + 1)$th row and $p$th column element.

Now we have,

\[
\max_{i,l,k} w_{il} |x_{ilk} - y_{ilk} r_{il}|
= \max_{i,l,k} \left| \sum_{p=1}^{n_1} w_{il} (t_{ilp} x_{p1k} - r_{il} \sum_{i=1}^{n_1} t_{ilp} x_{p1k}) \right|
= \max_{i,l,k} \left| \sum_{p=1}^{n_1} w_{il} (t_{ilp} - r_{il} \sum_{i=1}^{n_1} t_{ilp}) x_{p1k} \right|
= \max_{i,l,k} \left| \sum_{p=1}^{n_1} \gamma_{ilp} x_{p1k} \right|
\]
where \( \gamma_{ilp} = w_{il}(t_{ilp} - r_{il}) \).

Let the column vector \( X_k = (x_{11k}, x_{21k}, \ldots, x_{n1k})^T \) to be the cumulative production at level 1 during the time period 1 through \( k \). Hence the objective function (3.27) in Subsection 3.2.1, that is,

\[
\hat{Z}_{max}^a = \min \max_{i,l,k} w_{il} \left| x_{ilk} - y_{lk}r_{il} \right|
\]

at the time unit \( k \) over all parts, is transformed into matrix representation as follows:

\[
\text{minimize } \max_{i,l,k} w_{il} \left| x_{ilk} - y_{lk}r_{il} \right| = \min \max_k \| \Gamma X_k \|_1,
\]

where the norm \( \| . \|_1 \) is defined as

maximum \( \| a \|_1 = \max_i |a_i|, \ i = 1, 2, \ldots, n \), for a vector \( a = (a_1, a_2, \ldots, a_n) \).

Let the demand vector at level 1 be \( d = (d_1, d_2, \ldots, d_{n_1}) \) and the states in a schedule be \( X = (x_1, x_2, \ldots, x_{n_1}) \) with cardinality \( |X| = \sum_{i=1}^{n_1} x_i \) where \( x_i \) is the cumulative production of model \( i \), \( x_i \leq d_i \). Let \( e_i \) be the unit vector with \( n_1 \) entries all of which are zero except for a single 1 in the \( i \)th row.

Define \( \phi(X) \) to be the minimum of the maximum absolute deviation for all parts and models over all partial schedules of \( X \) and \( \| \Gamma X \|_1 \) is the maximum of the deviation of actual production from the ideal production over all parts and models when \( X \) is the amount of model produced.

The DP recursion for \( \phi(X) \), [75] is as follows:

\[
\phi(\emptyset) = \phi(X : X \equiv 0) = 0,
\]

\[
\phi(X) = \min_i \{ \max \{ \phi(X - e_i), \| \Gamma X \| , \} : i = 1, \ldots, n_1, \ x_i \geq 1 \}.
\]

For any state \( X \), it is observed that \( \phi(X) \geq 0 \) and \( \| \Gamma (X : X = d) \|_1 = 0 \).

Now we consider the objective function

\[
\sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_1} w_{il} (x_{ilk} - y_{lk}r_{il})^2.
\]

That is,

\[
\sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_1} w_{il} (x_{ilk} - y_{lk}r_{il})^2.
\]
\[= \sum_{k=1}^{D_1} \sum_{l=1}^{L} \sum_{i=1}^{n_l} w_{i,l} (t_{ilp}x_{p1k} - r_{il} \sum_{i=1}^{n_l} t_{ilp}x_{p1k})^2\]

\[= \sum_{k=1}^{D_1} (\|\Omega X_k\|_2^2)\]

where \(\Omega = w_{il}^2 \delta_{ilp}, \delta_{ilp} = t_{ilp} - r_{il} \sum_{i=1}^{n_l} t_{ilp}\).

The euclidean norm \(\|\cdot\|_2\) is defined as \(\|a\|_2 = \sqrt{\sum_{i=1}^{n} a_i^2}\) for a vector \(a = (a_1, \ldots, a_n)\).

Let \(\Phi(X)\) to be the minimum of the total square deviation respectively for all parts and models over all partial schedules of \(X\). The term \((\|\Omega X_k\|_2)^2\) is the sum of square deviations of actual production from the ideal production over all parts and models when \(X\) is the amount of model produced.

The DP recursion for \(\Phi(X)\), \([75]\) is

\[\Phi(\emptyset) = \Phi(X : X \equiv 0) = 0,\]

\[\Phi(X) = \min_{i} \{ \Phi(X - e_i) + (\|\Omega X\|_2)^2, i = 1, 2, \ldots, n_1, x_i \geq 1 \}\]

It is always true that \(\Phi(X) \geq 0\) and \(\Phi(X : X = d) = 0\) for any state \(X\).

In any state of \(X\), \(x_i\) can have any of the values \(0, 1, \ldots, d_i\). The number of states in the DP recursion is \(\prod_{i=1}^{n_1} (d_i + 1)\).

Any state \(X\) can be generated from \(n_1\) states.

The computation time for \(\|\Gamma X\|_1\) or \((\|\Omega X\|_2)^2\) is \(O(n_1 n), n = \sum_{m=1}^{L} n_m\). The space and time complexities of the DP procedures are \(O\left(\prod_{i=1}^{n_1} (d_i + 1)\right)\) and \(O\left(n_1 n \prod_{i=1}^{n_1} (d_i + 1)\right)\) respectively \([87]\).

The number of feasible schedules for any problem instance is \(\frac{D_1!}{d_1!d_2! \ldots d_{n_1}!}\). This is considerably larger than the number of states in the DP recursion. The in-
equality $\prod_{i=1}^{n_1} (d_i + 1) \leq \left( \frac{D_1 + n_1}{n_1} \right)^{n_1}$ shows that the DP algorithm is effective for small number of products even with large copies.

An excessive amount of time or that of space can be reduced by using some fast heuristics as a filter. The filter eliminates any states from DP’s state space that would lead to no optimality.

Two myopic heuristics exist for generating the filter [75]. One of the two heuristics shows that model $i$ becomes next model to be scheduled if that minimizes $\| \Gamma(X + e_i) \|_1$ and the other shows to minimize $\max \left\{ \| \Gamma(X + e_i) \|_1, \min_i \| \Gamma(X + e_i + e_i) \|_1 \right\}$

The DP algorithm progresses through the state space in the forward direction of increasing the cardinality as the procedure generates all states $X$ with $|X| = k$ before $|X| = k + 1$, $k = 1, 2, \ldots, D_1$ [75].

It is noteworthy that the output rate variation problem with a commutative aggregation function that aggregates deviations over all production cycles which is known as the symmetric output rate variation problem has been solved by the dynamic programming procedure [41].

### 4.3 Perfect Matching for Bottleneck PRVP

The bottleneck PRVP (i.e., MDJIT sequencing problem) was introduced and studied in the context of JIT car production systems [89], where the processor represents a mixed-model assembly line, and the $d_i$’s are the quantity of each type of car to be produced. Steiner and Yeomans [103] solved the absolute MDJIT problem $F_{\text{max}}^a$ under the constraints (3.3) – (3.6) by reducing it to an order preserving perfect matching problem via single machine scheduling release date/due date decision problem in a bipartite graph. A matching in a graph $G$ is a subset of edges such that no two edges are incident to the
same node and a perfect matching is incident to every vertex. A graph is 

bipartite if and only if it has no circuit of odd length [96]. The $j^{th}$ copy of the product $i$ is denoted by the pair $(i,j)$.

It is established in the literature that the perfect matching approach depends on the level curves $f_i(j - kr_i) = |j - kr_i|$, $j = 0, 1, \ldots, d_i$; $i = 1, 2, \ldots, n$; $k = 1, 2, \ldots, D$ and on the bound $B > 0$, known as bottleneck of the problem $F_{\text{max}}$. A copy $(i,j)$ is sequenced in a time unit $k$ in $[1, D]$ such that the level curves do not exceed $B$ [60].

For a given bound $B$, the earliest starting time $E(i, j)$ and the latest starting time $L(i, j)$ are respectively given by unique integers [25] such that

$$
\left(\frac{1}{r_i}\right) (j - B) - 1 \leq E(i, j) < \left(\frac{1}{r_i}\right) (j - B)
$$

and

$$
\left(\frac{1}{r_i}\right) [(j - 1) + B] - 1 < L(i, j) \leq \left(\frac{1}{r_i}\right) [(j - 1) + B]
$$

The sequencing times $E(i, j)$ and $L(i, j)$ for the weighted case are $\left\lfloor \frac{j - \frac{B}{w_i}}{r_i} \right\rfloor$ and $\left\lfloor \frac{j - 1 + \frac{B}{w_i}}{r_i} + 1 \right\rfloor$ respectively [26]. These sequencing times can be computed for each copy of each product in a one pass procedure with time complexity $O(D)$, $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, d_i$ [103]. The heavy weightage for particular copies of a product restricts the time window $[E(i, j), L(i, j)]$ and increases the separation of consecutive copies of that product in the production sequence.

The release date/due date decision problem may be represented as a perfect matching problem which is constructed in a $V_1 -$ convex bipartite graph $G = (V_1 \cup V_2, E')$ where

$V_1 = \{0, 1, \ldots, D - 1\}$ represents the set of sequencing times at starting,

$V_2 = \{(i, j) : i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i\}$, corresponds to the copy $j$ of product $i$, and

$E' = \{(k, (i, j)) : k \in [E(i, j), L(i, j)] \subseteq V_1\}$. 

57
The graph $G$ is said to be $V_1$-convex graph if $(i, j), (k, j) \in E'$ with $i < k \in V_1$ implies that $(l, j) \in E'$ for $i \leq l \leq k$.

A set $\mathcal{M} \subseteq E'$ in $G$ is matching if at most one $(k, (i, j)) \in \mathcal{M}$ is incident to the same vertex in $V_1 \cup V_2$. The matching $\mathcal{M}$ is perfect matching if every vertex in $V_1 \cup V_2$ is adjacent to some $(k, (i, j)) \in \mathcal{M}$. The perfect matching $\mathcal{M}$ is order-preserving with the property that the lower numbered copies of a model $i$ are sequenced to earlier sequencing times than the higher numbered copies. The perfect matching in $G$ is order-preserving for $B \leq 1$ since $E(i, j) \leq E(i, j + 1)$ and $L(i, j) \leq L(i, j + 1)$ [103].

Thus, in a $V_1-$ convex bipartite graph $G$, to find a feasible sequence in the release date/due date decision problem is similar as to find a perfect matching in bipartite graph $G$ with additional property that lower numbered copies of product are always matched to earlier starting times than higher numbered copies. This type of matching is called as order preserving matching [39, 53, 103].

**Theorem 4.2** The bottleneck PRVP, $F_{\text{max}}^a = \min \max_{i,k} |x_{ik} - kr_i|$ has a feasible solution if and only if the graph $G$ has a perfect matching.

In view of the convex nature of $G$, Steiner and Yeomans [103] suggested to use Golvers Earliest Due Date (EDD) algorithm [51] to check the existence of a perfect matching in $V_1-$ convex bipartite graph $G$. The EDD algorithm runs through the time instants $k = 1, 2, \ldots, D$ in order, and assigns to $k$ the part $(i, j)$ with smallest value of $L(i, j)$ among all the available parts such that $(k, (i, j)) \in E'$. The successive improvements in the efficiency of Glover’s algorithm can be found in [43].

Hall’s theorem is an important idea to establish a necessary and sufficient condition for the existence of a perfect matching in $G$ [25]. The Hall’s theorem states the following: a bipartite graph $G' = (V_1 \cup V_2, E')$ such that $|V_1| = |V_2|$
has a perfect matching if and only if \(|N(K)| \geq |K|, \forall K \subseteq V_1\) where \(N(K) = \{(i, j) : (i, j) \in V_2, \exists k \in K \text{ s.t.} (k, (i, j)) \in E'\}\). This can be applied to those sets \(K\) such that \(K\) is either an interval \(V_1\) in \(V_1\) or the neighborhood of \(V_1\) in \(V_1\) [25]. An order-preserving perfect matching in the graph \(G\) is similar to a feasible solution of the problem [103].

Taking \(V_1 = \{1, 2, \ldots, D\}\), Brauner and Crama [25] redefined the earliest and the latest starting times as \(E(i, j) = \left\lceil \frac{i-B}{r_i} \right\rceil\) and \(L(i, j) = \left\lfloor \frac{i-1+B}{r_i} + 1 \right\rfloor\) respectively. On using the Hall’s condition to the convex bipartite graph \(G\) associated with the absolute deviation objective \(F_{max}^a\), the following result is established therein [25]:

**Theorem 4.3** For all \(k_1, k_2 \in \{1, 2, \ldots, D\}\) with \(k_1 \leq k_2\), the bottleneck problem \(F_{max}^a\) with \(B < 1\) has a feasible solution if and only if
\[
\sum_{i=1}^{n} \max(0, \lfloor k_2 r_i + B \rfloor - \lceil (k_1 - 1) r_i - B \rceil) \geq k_2 - k_1 + 1
\]
and
\[
\sum_{i=1}^{n} \max(0, \lfloor k_2 r_i - B \rfloor - \lceil (k_1 - 1) r_i + B \rceil) \leq k_2 - k_1 + 1.
\]

Some algebraic properties of the problem are further characterized with result-oriented lemmas and conjectures [25]. The complexity of the problem \(F_{max}^a\) is not exactly known. Moreover, the problem is in Co-NP. For fixed \(n\), the optimization version of the problem can be reduced to integer linear programming with \(O(n)\) variables, in particular, the minimum value of \(B\) for the feasibility of the problem can be computed in polynomial time in \(O(\log D)\) steps [25].

### 4.3.1 Bounds for Bottleneck PRVP

In this subsection, we discuss about the upper bound and lower bound for the problem \(F_{max}^a\). Note that the words bound and bottleneck are used for
similar meanings hereafter. It is one established fact that an optimal sequence always exists when the deviation for each product is less than or equal to one [103]. That is, $B = 1$ is an upper bottleneck. From the point of quota apportionment method [7], that can be described as a version of the EDD algorithm applied with the bound, Brauner and Crama [25] have established that $B = 1 - \frac{1}{D}$ is also an upper bound which clearly implies that the upper bound $B = 1$ is not tight. The following theorem clearifies this fact.

**Theorem 4.4** The optimal value $B^*$ of the problem $F_{\text{max}}^a$ satisfies the inequality $B^* \leq 1 - \frac{1}{D}$.

Furthermore, $B = 1 - \frac{1}{2(n-1)}, n \geq 2$ is also an upper bottleneck [120]. It is observed that $D < 2(n-1)$ when $d_i = 1, \forall i$ with $n > 2$ but in most practical cases, $D \geq 2(n-1)$. Therefore, both the upper bounds have to be considered together. This implies that a bottleneck $B$ that yields an optimal sequence satisfies $B \leq 1 - \max \left\{ \frac{1}{D}, \frac{1}{2(n-1)} \right\}$ for $n \geq 2$.

A lower bound for bottleneck PRVP is $1 - r_{\text{max}}$, which is established as the tight lower bottleneck in [103]. The lower bound can not always be obtained. However, if it is achieved for an instance then it is an optimal bottleneck. The optimal bottleneck often coincides with the lower bound for small size instances [66]. The following theorem in [25] suggests slightly stronger bounds, where $\gcd$ stands for greatest common divisor.

**Theorem 4.5** If $\Delta_i = \frac{D}{\gcd(d_i, D)}, i = 1, 2, \ldots, n$, then optimal value $B^*$ of the problem $F_{\text{max}}^a$ satisfies the inequality $\frac{1}{\Delta_i} \left\lfloor \frac{\Delta_i}{2} \right\rfloor \leq B^* \leq 1 - \frac{1}{D}$.

The lower bound and hence the order-preserving product sequence is explained via an example in [103]: Assume $n = 5$ with respective demands $d_1 = 7, d_2 = 6, d_3 = 4, d_4 = 2, d_5 = 1$ such that the total demand is $D = 20$. The product rates are $r_1 = \frac{7}{20}, r_2 = \frac{6}{20}, r_3 = \frac{4}{20}, r_4 = \frac{2}{20}, r_5 = \frac{1}{20}$. It is
shown that the early and tardy starting times and the bipartite graph induced by the target value \( B = 0.65 = 1 - r_i = \frac{7}{20} = 1 - r_{\text{max}} \) for product 1. The EDD algorithm generates order-preserving product sequence (1-2-3-1-2-4-1-2-3-1-5-2-1-3-2-1-4-2-3-1), which implies that a copy of product 1 is produced, then a copy of product 2, then a copy of product 3 and so on. This solution is optimal, since this sequence is feasible and the target value is set at the lower bound of the problem.

For a given upper bottleneck, every instance \( d_1, d_2, \ldots, d_n \), with \( n \geq 2 \) has optimal sequence. However, the optimality is not guaranteed for smaller value, though better, of the given bottleneck. It is important and interesting to search instances with small optimal bottleneck. We note that no instance \((d_1, d_2, \ldots, d_n)\) with \( n \geq 2 \) has even feasible solution with \( B < \frac{1}{3} \) [30]. For any instance with single product \( n = 1 \), the upper bound is trivially 0.

For \( n = 2 \), there always exists optimal solution with the bound that does not exceed \( \frac{1}{2} \). When both \( d_1 \) and \( d_2 \) are odd, there exist two optimal sequences \( s_1 \) and \( s_2 \) for the bound \( B = \frac{1}{2} \). The difference in the two sequences is that if a copy of a product with demand \( d_1 \) or the other product with demand \( d_2 \) is in \( \frac{D}{2} \), then a copy of the other product is in \( \frac{D}{2} + 1 \) [73]. The optimal bottleneck for the problem \( F^a_{\text{max}} \) is less than \( B < \frac{1}{2} \) if and only if one of the demands \( d_1 \) or \( d_2 \) is odd and the other even [24]. A sequence with distances \( \left\lceil \frac{D}{d_1} \right\rceil \) or \( \left\lfloor \frac{D}{d_1} \right\rfloor \) for product 1 with demand \( d_1 \) and \( \left\lceil \frac{D}{d_2} \right\rceil \) or \( \left\lfloor \frac{D}{d_2} \right\rfloor \) for product 2 with demand \( D_2 \) is optimal. The two-product-type problem is briefly explained in the Subsection 4.3.4).

### 4.3.2 Improved Upper Bottleneck

The upper bound \( B = 1 \) is little reduced into the smaller bound \( B = 1 - \frac{1}{D} \). Since this bound is not tight, one has to look upon the tighter bounds as far as possible. In this context, we propose little improvement on the upper
bottleneck by taking the square root of $D$. That is, the new upper bottleneck appears to be $B = 1 - \frac{1}{\sqrt{D}}$. If $B^*$ denotes optimal value, then $B^* \leq 1 - \frac{1}{\sqrt{D}}$. Obviously, $B^* \leq 1 - \frac{1}{\sqrt{D}} \leq 1 - \frac{1}{D}$. Thus the bound $1 - \frac{1}{\sqrt{D}}$ is tighter one for the bottleneck PRVP which significantly reduces the deviation.

With this tighter upper bound, the theorem 4.4 and the theorem 4.5 respectively motivate to the following two conjectures:

**Conjecture 4.1** The optimal value $B^*$ of the problem $\text{F}^a_{\text{max}}$ satisfies the inequality $B^* \leq 1 - \frac{1}{\sqrt{D}}$.

**Conjecture 4.2** If $\Delta_i = \frac{\sqrt{D}}{\gcd(d_i, \sqrt{D})}, i = 1, 2, \ldots, n$; gcd = greatest common divisor, then optimal value $B^*$ of the problem $\text{F}^a_{\text{max}}$ satisfies the inequality $\frac{1}{\Delta_i} \left\lfloor \frac{\Delta_i}{2} \right\rfloor \leq B^* \leq 1 - \frac{1}{\sqrt{D}}$.

The conjecture 4.2 indicates the upper and lower bounds for the optimal value $B^*$ for the problem $\text{F}^a_{\text{max}}$. The bisection search can be performed in the interval $[1 - r_{\text{max}}, 1 - \frac{1}{\sqrt{D}}]$ which reduces the time complexity of the search. It can be observed that the new bound is efficiently workable for small instances in comparison of large instances. The further characterization of the bound remains as future research work.

### 4.3.3 Bisection Search

Any feasible solution is optimal if the bound $B$ is a minimum. Therefore, finding an optimal solution to the problem $\text{F}^a_{\text{max}}$ is finding a perfect matching with minimum $B$, which can be obtained by using a bisection search algorithm. An optimal solution has been obtained in [103] via bisection search in the interval $[1 - r_{\text{max}}, 1]$. Here we use $1 - \frac{1}{D}$ in place of 1. The bisection search algorithm runs in the interval $[1 - r_{\text{max}}, 1 - \frac{1}{D}]$, where $1 - r_{\text{max}}$ and $1 - \frac{1}{D}$ are the lower and the upper bounds respectively as in Subsection 4.3.1.
The minimum $B$ can be determined in $O(\log D)$ time \cite{103}. Since sequencing times are determined in $O(D)$ time and minimum $B$ in $O(\log D)$ time, we can conclude that an optimal solution can be determined in $O(D\log D)$ time. The bisection search algorithm can find an optimal sequence in $O(D\log(D\phi w_{max}))$, where $\phi$ is an integer and $w_{max} = \max\{w_i\}, i = 1, 2, \ldots, n$, for the weighted case \cite{105}.

Though a number of efforts has been performed to solve the problem with polynomial complexity on the input size of the demands, the exact complexity still remains open. Since infeasibility of an instance can be checked in $O(n\log D)$ time, the problem is Co-NP. But it has not been decided whether the problem is Co-NP-complete or polynomially solvable \cite{25}. From the analysis of existing work-in-progress in the literature, solution of this problem with polynomial time seems unlikely to exist \cite{60}.

It is noteworthy that the perfect matching with a bisection search determines optimal sequence in less time than the bottleneck assignment approach does. This shows that the perfect matching with bisection search is more efficient though it is pseudo-polynomial.

### 4.3.4 Problem with Two Products

The bottleneck product rate variation problem with the two products ($n = 2$) is studied in \cite{25} with the demands $d_1$ and $d_2$ such that the total demand is $D = d_1 + d_2$. The ideal production rates in this case are $r_1$ and $r_2 = 1 - r_1$ such that $r_1 = \frac{d_1}{D}$ and $r_2 = \frac{d_2}{D}$. The optimization version of the two-product type problem is to find a $2 \times D$ matrix $X = (x_{ik})$ which minimizes the maximum deviation under the related constraints \cite{25}. The mathematical model of this problem is formulated as the following:

$$\max_{1 \leq k \leq D} (|x_{1k} - kr_1|, |x_{2k} - kr_2|)$$

(4.10)
subject to
\[ x_{1k} + x_{2k} = k, \quad k = 1, 2, \ldots, D \]  
(4.11)  
\[ x_{i(k-1)} \leq x_{ik}, \quad i = 1, 2; \quad k = 2, 3, \ldots, D \]  
(4.12)  
\[ x_{i0} = 0; \quad x_{iD} = d_i, \quad i = 1, 2 \]  
(4.13)  
\[ x_{ik} \in \mathbb{Z}^+, \quad i = 1, 2, \ldots, D \]  
(4.14)

The optimal solution of this problem is obtained and computed in polynomial time as follows: A matrix \( X \) defined by \( x_{1k} = [kr_1] \) and \( x_{2k} = k - [kr_1], k = 1, 2, \ldots, D \) is an optimal solution of the two-product-typed maximum deviation sequencing objective (4.10). Consequently at every instant \( k \), this fact permits to determine efficiently which product should be produced at time \( k \). And the optimal value \( B^* \) is computed by the formula \( B^* = \frac{1}{\Delta} \lfloor \frac{\Delta}{2} \rfloor \), where \( \Delta = \frac{D}{\gcd(d_1, D)} = \frac{D}{\gcd(d_2, D)}, \gcd \) stands for greatest common divisor. It is noteworthy that the matrix \( X \) defined as above actually minimizes the deviation \( x_{ik} - kr_i \) for all \( k \) and \( i \). Therefore, it is optimal for the bottleneck PRVP and the total PRVP as well, with any convex and symmetric penalty function \( F_i \). This solution procedure is formalized by the following two theorems [25].

**Theorem 4.6** The matrix \( X \) defined by \( x_{1k} = [kr_1] \) and \( x_{2k} = k - [kr_1], k = 1, 2, \ldots, D \) is an optimal solution of the two-product-type bottleneck PRVP.

**Theorem 4.7** The optimal value \( B^* \) of the objective function of the two-product-type bottleneck PRVP is \( B^* = \frac{1}{\Delta} \lfloor \frac{\Delta}{2} \rfloor \) where \( \Delta = \frac{D}{\gcd(d_1, D)} = \frac{D}{\gcd(d_2, D)} \).

### 4.3.5 Problem with Small Bottlenecks

The extensive analysis with some conjectures has been observed in [25] for the instances of bottleneck PRVP with \( B^* \leq \frac{1}{2} \) and all instances has been identified with small deviation, i.e., with \( B^* < \frac{1}{2} \) with \( n \leq 6 \). An instance of
the problem $F^a_{\text{max}}$ with demands $d_1, d_2, \ldots, d_n$ is called \textit{standard} if $d_1 \leq d_2 \leq \ldots \leq d_n$ and $\gcd(d_1, d_2, \ldots, d_n, D) = 1$.

There exists only one instance with the bound less than $\frac{1}{2}$ for $n \geq 3$. A conjecture, namely \textit{small deviation conjecture (SDC)}, has been proposed in [25] which states that only the standard instance $(1, 2, 2^2, \ldots, 2^{i-1}), i = 1, 2, \ldots, n$ has optimal bottleneck less than $\frac{1}{2}$ and vice versa.

\textbf{Conjecture 4.3} [25] For $n \geq 3$, a standard instance $d_1, d_2, \ldots, d_n$ of the bottleneck PRVP $F^a_{\text{max}}$ has optimal value $B^* = \frac{2^{n-1} - 1}{2^{n-1}} < \frac{1}{2}$ if and only if $d_i = 2^{i-1}, i = 1, 2, \ldots, n$.

The conjecture is shown to be true with two proofs. The first proof shows that the SDC is a consequence of the Fraenkel’s conjecture for symmetric case. The symmetric case of the Fraenkel’s conjecture has been proved using the concept of balanced word [24]. The second proof is via geometric representation [68]. The remaining part of this subsection is the brief explanation of Fraenkel’s conjecture with balanced word and the geometric representation for SDC.

The concept of balanced word is one of the good solutions for problems of finding a fair sequence that allocates capacity to competing demands such that each demand receives a share of the capacity that is approximately proportional to its priority, over any time period. The idea of fair sequences occurs in many different areas such as scheduling mixed-model just-in-time assembly lines, inventory management, and controlling asynchronous transfer mode networks. A good overview of fair sequences in different domains and an extensive discussion on the results for multiple related problems such as the product rate variation problem, generalized pinwheel scheduling, the hard real-time periodic scheduling problem, stride scheduling, minimizing response time variability and peer-to-peer fair scheduling can be found in [69].

65
On a finite set \( \{1, 2, \ldots, n\} \), a \( \delta \)-balanced word is an infinite sequence \( S = S_1S_2\ldots \) with \( S_i \in \{1, 2, \ldots, n\} \) such that every two subsequences of equal length consist of only those letters \( S_i \) whose number of occurrences in each subsequence differs by at most a positive integer \( \delta \) [127]. An 1-balanced word is known as a balanced word. Consider a finite word \( s = S_1S_2\ldots S_D \) on \( \{1, 2, \ldots, n\} \) of length \( D \) with \( d_i \) occurrences of a letter \( S_i \in \{1, 2, \ldots, n\} \) and \( r_i = \frac{d_i}{D} \), the rate of letter \( S_i \) with \( r_1 \leq r_2 \leq \ldots \leq r_n \). The word \( S \) is symmetric if \( s = s^R \), where \( s^R = S_D\ldots S_2S_1 \) is the mirror reflection of \( S \). An infinite word \( s \) is periodic if \( S = ss\ldots \) for some \( s \).

Consider a finite word (finite sequence) \( s \) with the length \( D \) and suppose that each letter \( S_i \in \{1, 2, \ldots, n\} \) consists of \( d_i \) copies such that \( D = \sum_{i=1}^{n} d_i \). Let \( B \) be the bound for the finite word \( s \). Jost [57] established the fact that any infinite periodic word \( S \) with the period \( s \) is 1-balanced, 2-balanced and 3-balanced if the bottleneck is \( B < \frac{1}{2} \), \( B < \frac{3}{4} \) and \( B < 1 \) respectively. It is observed that the set of such infinite periodic words with \( B < \frac{1}{2} \), \( B < \frac{3}{4} \) or \( B < 1 \) is properly contained in the set of 1-balanced, 2-balanced or 3-balanced words respectively [33]. Note that the 1-balanced words are unlikely to exist for most cases. An optimal sequence for the problem \( F_{max}^{a} \) exists in the set of all 3-balanced words. However, it remains unresolved whether there always exists a 2-balanced word that is optimal for the problem \( F_{max}^{a} \). The challenging problem of balanced words in practice is to construct an infinite periodic sequence over a finite set of letters with given rates and distributed as evenly as possible [31, 33].

Induction on \( n \) is used to show that a periodic symmetric balanced word on \( n \geq 3 \) letters with the ratios \( r_1 < r_2 < \ldots < r_n \) exists if and only if \( r_i = \frac{2^{i-1}}{2^n - 1} \). This is known as Fraenkel’s conjecture for symmetric case [120]. A periodic balanced word can be constructed with the ratios in Fraenkel’s conjecture. A periodic balanced word for \( n = 2 \) is 212. A new letter 3 is inserted between every consecutive letters as well as at the end and in the beginning.
to construct a periodic balanced word for \( n = 3 \). The word is 3231323. The cardinality of the word is \( 2n + 1 \). The procedure is followed for higher numbered letters; however, the necessity is challenging. The conjecture in terms of balanced word is true for \( n \leq 6 \) [122]. It has been shown to be true for \( n = 3 \) in [121] and for \( n = 4 \) in [2]. The conjecture is still open for \( n > 6 \). However, the symmetric case of the conjecture in terms of balanced word has been proved by induction on \( n \) [24].

A solution \( x_{ik}, i = 1, 2, \ldots, n; k \geq 1, 2, \ldots, D \) of the problem \( F_{\text{max}}^a \) with the bound \( B < \frac{1}{2} \) generates a periodic symmetric balanced word [24]. The truthness of the Fraenkel’s conjecture implies the truthness of SDC. Thus, the small deviation conjecture is a consequence of the symmetric case of the Fraenkel’s conjecture.

The SDC is valid for \( 3 \leq n \leq 6 \) according to [25]. However, Kubiak [68] proved the SDC for any \( n \geq 3 \) by showing that if \( B < \frac{1}{2} \) then all copies of all products must be sequenced in their ideal positions \( \left[ \frac{2j-1}{2r_i} \right] \) for copy \( j \) of product \( i, j = 1, 2, \ldots, d_i \). His geometric proof relied on the ideal positions and on natural symmetry of regular product polygons inscribed in a circle of circumference \( D \) such that each polygon corresponds to a different model having \( d_i \) corners (the number \( \frac{2j-1}{2r_i} \) is referred as the ideal corner for the copy \( j \) of the product \( i \)) for the product \( i \) at \( \left[ \frac{2j-1}{2r_i} \right], i = 1, 2, \ldots, n \) points on the perimeter of the circle [68]. The proof shows that if the bound is \( B < \frac{1}{2} \), then all copies \( (i, j) \) of model \( i, i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i \) must be sequenced in the ideal positions. This is possible if \( d_n \) is divisible by each \( d_i, i = 1, 2, \ldots, n - 1 \). As a result, \( n \) demands are the first \( n \) non-negative powers of 2 [68].

For \( n = 2 \), Kubiak further proved that there are infinitely many instances with optimal value less than \( \frac{1}{2} \) in the following theorem.

\textbf{Theorem 4.8} For \( n = 2 \), the optimal value of the problem \( F_{\text{max}}^a \) is less than...
if and only if one of the demands $d_1$ or $d_2$ is even and the other is odd.

Kubiak further presented a complete characterization of instances with small deviations (i.e., with less than $\frac{1}{2}$) for two products, and consequently characterized the instances with small deviations for any number of products. Finally, he exploited these results to prove special cases of the well-known Fraenkel’s conjecture.

All in the above, we discused on the bottleneck PRVP with absolute deviation objective $F_{a\max}$. In the next section, we briefly discuss on the bottleneck PRVP with general index, as an extension of bottleneck PRVP with square deviation [60]. The lower and the upper bottlenecks to the problem $F_{\max}^s$ have been established and a pseudo-polynomial optimization algorithm, i.e., a modified perfect matching with a bisection search, has been obtained.

### 4.4 Bottleneck PRVP with General Index

The non-linear integer programming for the bottleneck PRVP with general index is the problem $F_m = \min \max_{i,k} |x_{ik} - kr_i|^m$ under the constraints (3.3) to (3.6), where $m$ is positive integer. The problem $F_m$ is solved by reducing to an order-preserving perfect matching problem via single machine scheduling release/due date decision problem. The order-preserving perfect matching gives a feasible solution to the problem. A bisection search in the interval of lower and upper bottlenecks obtains an optimal solution. This procedure is a modified version of the method that appeared in [103].

**Construction of permissible time window**

It has been established that the perfect matching relies on the level curves $f_i(j - kr_i) = |j - kr_i|^m$, $j = 0, 1, \ldots, d_i; i = 1, 2, \ldots, n; k = 1, 2, \ldots, D$ and the bound $B > 0$. The level curves are sketched over the interval $[1, D]$ that covers the planning horizon. The bottleneck is taken as a line parallel
to the horizontal axis. The points where the bottleneck crosses the level curves are useful to find the sequencing times. A copy \((i, j)\) is sequenced in a time unit \(k\) in \([1, D]\) such that the level curves do not exceed the bound \(B\). These type of times are referred as sequencing times, which depend on \(B\) and \(m\). The bottleneck \(B < 1\) crosses the level curves with different values for \(m\) at different points showing that there may exist instances with different sequencing times for different objective functions of the problem [60].

For a given bound \(B\), let \(E_m(i, j)\) and \(L_m(i, j)\) be the earliest and the latest sequencing times of \((i, j)\) respectively to the problem \(F_m\). The earliest sequencing time \(E_m(i, j)\) is the unique integer in \([1, D]\) such that when \((i, j)\) is sequenced at \(E_m(i, j) - 1\), the level curves exceed \(B\) but do not exceed when sequenced at \(E_m(i, j)\). Similarly, the latest sequencing time \(L_m(i, j)\) is the unique integer such that when \((i, (j - 1))\) is sequenced at \(L_m(i, j) - 1\), the level curves do not exceed \(B\) but exceed when sequenced at \(L_m(i, j)\).

\[
\begin{align*}
\text{Figure 4.1: Level curves for } & d_1 = 4; B = 1 \text{ to problem } F_{\max}^a \\
\end{align*}
\]

**Theorem 4.9** [60] For a given bottleneck \(B\), the earliest sequencing time \(E_m(i, j)\) and the latest sequencing time \(L_m(i, j)\) are \(E_m(i, j) = \left\lceil \frac{i - \sqrt{B/r_i}}{r_i} \right\rceil\) and \(L_m(i, j) = \left\lfloor \frac{i - 1 + \sqrt{B/r_i}}{r_i} + 1 \right\rfloor\), \(i = 1, 2, \ldots, n; j = 0, 1, \ldots, d_i\) respectively.

The immediate corollary for the weighted case of the problem is the following.
Corollary 4.1 [60] The earliest and latest sequencing times for the weighted case of the problem $F_m$ are $E_m(i, j) = \left\lfloor \frac{j - \sqrt{\frac{B}{w_i}}}{r_i} \right\rfloor$ and $L_m(i, j) = \left\lceil \frac{j - 1 + \sqrt{\frac{B}{w_i}}}{r_i} \right\rceil + 1$, $i = 1, \ldots, n; j = 0, 1, \ldots, d_i$, where $w_i$ be the weight for model $i$, $i = 1, 2, \ldots, n$.

An interval $T_m = [E_m(i, j), L_m(i, j)]$ is defined as a time window in which $(i, j)$ can be produced that permits to form a feasible sequence. Thus, the time window $T_m$ is the permissible time window for the production of $(i, j)$.

Corollary 4.2 If $(i, j)$ be sequenced within the time window $T_m = [E_m(i, j), L_m(i, j)]$, the level curves do not exceed $B$.

For a given bound $B$, the earliest and the latest sequencing times $E_m(i, j)$ and $L_m(i, j)$ can be calculated for each $(i, j)$, $i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i$. They are calculated in $O(D)$ time [103], and hence the time window $T_m$ can be constructed in $O(D)$ time. Next, we explain the existence of perfect matching, lower and upper bottlenecks, and modified bisection search for the problem $F_m$.

Existence of perfect matching

Consider the set of sequencing times as $V_1 = \{1, 2, \ldots, D\}$ and the set of $j^{th}$ copy of model $i$ as $V_2 = \{(i, j) : i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i\}$. A $V_1$-convex bipartite graph $G = (V_1 \cup V_2, \mathcal{E})$ is constructed by sequencing $(i, j)$ within $T_m$, where $\mathcal{E} = \{(k, (i, j)) : k \in T_m\}$. The earliest due date (EDD) algorithm that matches each $k \in V_1$ to the unmatched $(i, j)$ with the smallest $L_m(i, j)$ and $(k, (i, j)) \in \mathcal{E}$ finds a perfect matching. This is a modified version of Glover’s EDD rule to find a maximum matching in a $V_1$-convex bipartite graph $G$ [103]. Glover’s EDD rule states that if unmatched $(i, j)$ does not
exist for $k$, the algorithm moves to $k + 1$. However, the modified algorithm stops in such a situation. We refer [60] for detail.

The necessary and sufficient condition for the existence of a perfect matching in $G$ is provided by Hall’s theorem, which states that a perfect matching in $G = (V_1 \cup V_2, \mathcal{E})$ with $|V_1| = |V_2|$ exists if and only if $|N(K)| \geq K$ for all $K \subseteq V_1$, where $N(K) = \{(i, j) : (i, j) \in V_2, \exists k \in K \text{ such that } (k, (i, j) \in \mathcal{E})\}$. Hall’s condition can be modified for the existence of a perfect matching in the case of $\mathcal{G}$ formed by any instance $(d_1, d_2, \ldots, d_n; B)$ of the problem $F_m$. The modification is that $K$ is either an interval in $V_1$ or the neighborhood of an interval in $V_1$ [25]. The existence of a perfect matching in $G$ depends on $B$. A perfect matching in $G$ exists if $B$ satisfies the inequalities in Theorem 4.10, which is a certificate for the existence of a perfect matching.

**Theorem 4.10** [60] A perfect matching in the $V_1$-convex bipartite graph $G = (V_1 \cup V_2, \mathcal{E})$ exists if and only if, for all $k_1, k_2 \in V_1$ with $k_1 \leq k_2$ and $[E_m(i, j), L_m(i, j)] \cap [k_1, k_2] \neq \emptyset$, $B$ satisfies the inequalities

$$\sum_{i=1}^{n}(\left\lfloor k_2 r_i + \frac{\sqrt{B}}{w_i} \right\rfloor - \left\lfloor (k_1 - 1)r_i - \frac{\sqrt{B}}{w_i} \right\rfloor) \geq k_2 - k_1 + 1$$

and

$$\sum_{i=1}^{n}(\left\lfloor k_2 r_i - \frac{\sqrt{B}}{w_i} \right\rfloor - \left\lfloor (k_1 - 1)r_i + \frac{\sqrt{B}}{w_i} \right\rfloor) \leq k_2 - k_1 + 1.$$

The certificate for the existence of a perfect matching in the weighted case is as follows.

**Corollary 4.3** A perfect matching in the $V_1$-convex bipartite graph formed by the weighted case of problem $F_m$ exists if and only if, $[E_m(i, j), L_m(i, j)] \cap [k_1, k_2] \neq \emptyset$ and for all $k_1 \leq k_2$ in $V_1$, $B$ satisfies the inequalities

$$\sum_{i=1}^{n}(\left\lfloor k_2 r_i + \frac{B}{w_i} \right\rfloor - \left\lfloor (k_1 - 1)r_i - \frac{B}{w_i} \right\rfloor) \geq k_2 - k_1 + 1$$

and

$$\sum_{i=1}^{n}(\left\lfloor k_2 r_i - \frac{B}{w_i} \right\rfloor - \left\lfloor (k_1 - 1)r_i + \frac{B}{w_i} \right\rfloor) \leq k_2 - k_1 + 1.$$

The modified EDD algorithm sequences the lower numbered copies of a model to earlier sequencing times than the higher numbered copies, which leads
the perfect matching to be order-preserving. That is, a perfect matching in \( G \) is order-preserving [60]. The feasible sequence has its relation with order-preserving perfect matching. An order-preserving perfect matching in \( G \) is analogous to a feasible solution to the problem \( F_m \). Any instance \((d_1, d_2, \ldots, d_n; B)\) of the problem \( F_m \) has a feasible sequence if and only if the \( V_1\)-convex bipartite graph formed by the instance has an order-preserving perfect matching.

On defining the lower bottleneck \((1-r_{\text{max}})^m, r_{\text{max}} = \max \{r_i\}, i = 1, 2, \ldots, n;\) and the upper bottleneck \((1 - \frac{1}{D})^m\) for \( F_m \), the bisection search for \( F_{\text{max}} \) has been modified for the problem \( F_m \) [60]. The modified bisection search that runs in the interval \([(1-r_{\text{max}})^m, (1 - \frac{1}{D})^m]\) determines the minimum \( B \) in \( O(\log D) \) time. Note that the problem \( F_m \) is in Co-NP but is still unresolved whether it is Co-NP-complete or polynomially solvable [25]. The observation of the input size \( O(n \log D) \) involving \( nD \) variables and \( O(nD) \) constraints in the model indicate that the existence of a polynomial algorithm seems far from trivial. It is noteworthy that the bisection search can be remodified in the interval with the new upper bottleneck that we have proposed in Subsection 4.3.2.

4.5 Assignment Approach for Total PRVP

The total PRVP, \( F_{\text{sum}} = \min \sum_{k=1}^{D} \sum_{i=1}^{n} F_i(x_{ik} - kr_i), \) under the constarints (3.3) to (3.6) is reduced into an equivalent assignment problem and solved with pseudo-polynomial complexity [76, 74]. They assumed \( F_i \) as a unimodal convex symmetric penalty function satisfying \( F_i(0) = 0 \) and \( F_i(y) > 0, y \neq 0, i = 1, 2, \ldots, n. \) The core idea of the assignment algorithm is calculation of the ideal position and the assignment costs:

The ideal position for each product \( i \) is computed by the formula \( Z_{i}^{*} = \)
\[ \left\lfloor \frac{2^{i-1} - 1}{2r_i} \right\rfloor, n = 1, 2, \ldots, n \] and \( j = 1, 2, \ldots, d_i \), which is the ceiling of the unique crossing point of \((i, j)\) satisfying \( F_i(j - kr_i) = F_i(j - 1 - kr_i) \). The ceiling \( Z_j^{i*} \) of the unique crossing point is important for the calculation of the assignment costs. When \((i, j)\) is sequenced at the time unit with the value equal to \( \left\lfloor \frac{2^{j-1} - 1}{2r_i} \right\rfloor \), the \( j^{th} \) copy of product \( i \) contributes the cost \( \inf_j F_i(j - kr_i) \) to the total cost. Therefore, \( \left\lfloor \frac{2^{j-1} - 1}{2r_i} \right\rfloor \) is called the ideal position (See [56] for detail). Moreover, we have the following theorem in [71, 74]:

**Theorem 4.11** A solution \( \left( \left\lfloor \frac{2^{x+1} - 1}{2r_i} \right\rfloor, \left\lfloor \frac{2^{x+2} - 1}{2r_i} \right\rfloor, \ldots, \left\lfloor \frac{2^{x+d_i} - 1}{2r_i} \right\rfloor \right) \) is optimal to total PRVP with the objective function \( \sum_{k=1}^{D} \inf_j F_i(j - kr_i) \).

Next, we define **assignment cost** as follows: If \((i, j)\) is not sequenced at the ideal position, then clearly there occur some additional costs. Let \( C_{jk}^i \geq 0 \) be the cost of assigning \((i, j)\) to the period \( k \). If \( k < Z_j^{i*} \), then \( j^{th} \) copy of product \( i \) is produced too early and hence excess inventory costs are incurred in the periods from \( l = k \) to \( l = Z_j^{i*} - 1 \). If \( k = Z_j^{i*} \), then \( j^{th} \) copy of product \( i \) is produced at ideal position and hence \( C_{jk}^i = 0 \). If \( k > Z_j^{i*} \), then \( j^{th} \) copy of product \( i \) is produced too lately and hence the excess shortage costs are incurred in periods from \( l = Z_j^{i*} \) to \( l = k - 1 \). Thus, the sequencing cost is computed by the formula:

\[
C_{jk}^i = \begin{cases} 
Z_j^{i*} - 1, & \text{if } k < Z_j^{i*} \\
\sum_{l=k}^{k-1} \Psi_{jl}^i, & \text{if } k = Z_j^{i*} \\
\sum_{l=Z_j^{i*}}^{k-1} \Psi_{jl}^i, & \text{if } k > Z_j^{i*} 
\end{cases}
\]

\[
\Psi_{jl}^i = |F_i(j - lr_i) - F_i(j - 1 - lr_i)| = \begin{cases} 
F_i(j - lr_i) - F_i(j - 1 - lr_i), & \text{if } l < Z_j^{i*} \\
F_i(j - 1 - lr_i) - F_i(j - lr_i), & \text{if } l \geq Z_j^{i*} 
\end{cases}
\]

and \((i, j) \in I = \{(i, j) : i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i\}, l = 1, 2, \ldots, D.\)
Table 4.1: Ideal positions for \( d_1 = 3, d_2 = 4, d_3 = 5; D = 12 \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( j )</th>
<th>( \frac{2j-1}{2r_i} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>11</td>
</tr>
</tbody>
</table>

We consider an example to illustrate the ideal position and assignment costs:

**Example 4.1:** Consider the three products \( n = 3 \) with the demands \( d_1 = 3, d_2 = 4, d_3 = 5 \) such that the total demand is \( D = 12 \).

The ideal positions for each product \( i \) to be produced are calculated in Table 4.1. The costs \( \Psi_{jl}^i \) for \( F_{sum}^a \) are calculated in Table 4.2, and the additional sequencing costs \( C_{jk}^i \) are calculated in Table 4.3.

If \((i,j)\) is sequenced at the ideal position for all \( i \) and \( j \), no copy competes with the other. That is, the copies are competition free. Sequencing all copies of all models at their ideal positions minimizes the PRVP and the sequence obtained with this is obviously optimal. But the competition occurs for the positions in general, that leads to infeasibility [71].

In Example 4.1, the positions 4, 5, 8, 9, 10 are competition-free. The copies \((1,1)\), \((2,1)\) and \((3,1)\) compete for position 2, copies \((1,2)\) and \((3,3)\) for
Table 4.1: Ideal positions for $d_1 = 3, d_2 = 4, d_3 = 5; D = 12$

$$\begin{array}{lcccccccccccc}
  & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
 1 1 & 0.5 & 0 & 0.5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 2 & 1 & 1 & 1 & 1 & 0.5 & 0 & 0.5 & 1 & 1 & 1 & 1 & 1 \\
 1 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.5 & 0 & 0.5 & 1 \\
 2 1 & 0.3 & 0.3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 2 2 & 1 & 1 & 0.9 & 0.3 & 0.4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 2 3 & 1 & 1 & 1 & 1 & 1 & 0.9 & 0.3 & 0.4 & 1 & 1 & 1 & 1 \\
 2 4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.9 & 0.3 & 0.4 & 1 & 1 \\
 3 1 & 0.2 & 0.6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 3 2 & 1 & 1 & 0.6 & 0.3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 3 3 & 1 & 1 & 1 & 1 & 1 & 0.8 & 0.2 & 1 & 1 & 1 & 1 & 1 \\
 3 4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.3 & 0.5 & 1 & 1 & 1 \\
 3 5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.6 & 0.2 & 1 & 1 \\
\end{array}$$

position 6, copies $(2, 4)$ and $(3, 5)$ for position 11. Note that the positions 1, 3, 7, 12 are left unassigned with any $(i, j)$. Higher priority is given to $j$ over $j'$ whenever $j < j'$ to avoid competition and $(i, j)$ is assigned to a position $k$, $k \neq \lceil \frac{2j-1}{2r_i} \rceil$ [74]. Additional cost is incurred if $(i, j)$ is sequenced at the position other than the ideal position. We assume that $(2, 1)$ is assigned to 2 and $(3, 1)$ to 3; $(1, 2)$ to (6) and $(3, 3)$ to 7; $(2, 4)$ to 11 and $(3, 5)$ to 12. This new assignment generates a sequence $2 - 3 - 1 - 3 - 2 - 3 - 1 - 2 - 3 - 1 - 3 - 2$. The sequence means a copy of model 2 is produced during the first time unit, a copy of model 3 in the second time unit, a copy of model 1 in the third time unit and so on upto the complete production to meet the demand in sequence.

On defining the production variable

$$x^i_{jk} = \begin{cases} 
1 & \text{if } (i, j) \text{ is assigned to period } k \\
0 & \text{otherwise}
\end{cases}$$

75
Table 4.3: The additional costs $C^i_{jk}$ for $F^a_{sum}$ with $d_1 = 3, d_2 = 4, d_3 = 5; D = 12$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>1.5</td>
<td>2.5</td>
<td>3.5</td>
<td>4.5</td>
<td>5.5</td>
<td>6.5</td>
<td>7.5</td>
<td>8.5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4.5</td>
<td>3.5</td>
<td>2.5</td>
<td>1.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>1.5</td>
<td>2.5</td>
<td>3.5</td>
<td>4.5</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>8.5</td>
<td>7.5</td>
<td>6.5</td>
<td>5.5</td>
<td>4.5</td>
<td>3.5</td>
<td>2.5</td>
<td>1.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.3</td>
<td>0</td>
<td>0.3</td>
<td>1.3</td>
<td>2.3</td>
<td>3.3</td>
<td>4.3</td>
<td>5.3</td>
<td>6.5</td>
<td>7.3</td>
<td>8.3</td>
<td>9.3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3.2</td>
<td>2.2</td>
<td>1.2</td>
<td>0.3</td>
<td>0</td>
<td>0.4</td>
<td>1.4</td>
<td>2.4</td>
<td>3.4</td>
<td>4.4</td>
<td>5.4</td>
<td>6.4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6.2</td>
<td>5.2</td>
<td>4.2</td>
<td>2.7</td>
<td>2.4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0.4</td>
<td>1.4</td>
<td>2.4</td>
<td>3.4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>9.2</td>
<td>8.2</td>
<td>7.2</td>
<td>6.2</td>
<td>5.2</td>
<td>4.2</td>
<td>3.2</td>
<td>2.2</td>
<td>1.9</td>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.2</td>
<td>0</td>
<td>0.6</td>
<td>1.6</td>
<td>2.6</td>
<td>3.6</td>
<td>4.6</td>
<td>5.6</td>
<td>6.6</td>
<td>7.6</td>
<td>8.6</td>
<td>9.6</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2.6</td>
<td>1.6</td>
<td>0.6</td>
<td>0</td>
<td>0.3</td>
<td>1.3</td>
<td>2.3</td>
<td>3.3</td>
<td>4.3</td>
<td>5.3</td>
<td>6.3</td>
<td>7.3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4.8</td>
<td>3.8</td>
<td>2.8</td>
<td>1.8</td>
<td>0.8</td>
<td>0</td>
<td>0.2</td>
<td>1.2</td>
<td>2.2</td>
<td>3.2</td>
<td>4.2</td>
<td>5.2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>7.3</td>
<td>6.3</td>
<td>5.3</td>
<td>4.3</td>
<td>3.3</td>
<td>2.3</td>
<td>1.3</td>
<td>0.3</td>
<td>0</td>
<td>0.5</td>
<td>1.5</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>9.6</td>
<td>8.6</td>
<td>7.6</td>
<td>6.6</td>
<td>5.6</td>
<td>4.6</td>
<td>3.6</td>
<td>2.6</td>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The total PRVP is formulated as the following equivalent assignment problem [71, 74]:

**Theorem 4.12** An optimal solution to minimize the problem $F_{sum}$ subject to the constraints (3.3) to (3.6) can be obtained from any optimal solution of the assignment problem:

\[
\text{minimize} \quad F^A_{sum} = \sum_{k=1}^{D} \sum_{(i,j) \in I} C^i_{jk} x^i_{jk} \quad (4.15)
\]

subject to \( \sum_{(i,j) \in I} x^i_{jk} = 1, \quad k = 1, 2, \ldots, D \quad (4.16) \)

\[
\sum_{k=1}^{D} x^i_{jk} = 1, \quad (i, j) \in I \quad (4.17)
\]

\[
x^i_{jk} = 0 \text{ or } 1, \quad k = 1, 2, \ldots, D, \quad (i, j) \in I \quad (4.18)
\]

76
Note that $F^A_{sum}$ stands for the assignment formulation of the problem $F_{sum}$. The assignment problem is efficiently solvable; viz., the problem with $2D$ nodes can be solved in $O(D^3)$ time [96]. More precisely, a primal-dual algorithm, called Hungerian method [77], takes $O(D^3)$ arithmetic operations to solve the assignment problem with $2D$ nodes. There are $D^2$ values for $\Psi^i_{jk}$ and the $D^2$ values for $C^i_{jk}$ to calculate, each taking $O(D)$ steps which shows that the computation of the assignment costs takes $O(D^3)$ steps. A parallel algorithm is developed which can efficiently solve assignment problems with 900 million variables [3].

It is proved in [33] that the assignment problem cannot be solved at optimality only under the constraints (4.16) to (4.18); however, needs another constraint, not of assignment type: if $(i, j, k)$ and $(i, j', k')$ are feasible schedules with $k < k'$, then $j < j'$, i.e., lower indices copies are produced earlier, which imposes an order on copies of a product. This constraint is essential as it ties up the copy $j$ of a product with the $j^{th}$ ideal position for the product. Looking on the efficiency, the order of the copies can be reordered for the optimality and the latter copies can be produced earlier in the production sequence. The assignment approach for total PRVP is applicable to any $lp$-norm; and in particular, to $l_{\infty}$-norm minimizing bottleneck PRVP objective [33]. Some structural properties of the min-sum just-in-time level schedule problem are described in [1]. In the following subsection, we describe the feasibility and optimality of the total PRVP.

### 4.5.1 Feasibility and Optimality

The systematic transformation of the total PRVP into assignment problem is given in [74], where the feasibility and the optimality of the problem are subsequently proved.

Let $\mathcal{Y} = \{(i, j), k): i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i; k = 1, 2, \ldots, D\}$ be the
set of the assignment of \((i, j)\) to the period \(k\). A set \(Y \subseteq \mathcal{Y}\) is said to be \(Y\)-feasible if the following constraints hold [33, 74].

\(c_1\). Exactly one copy is produced at one time unit. That is, for each \(k, k = 1, 2, \ldots, D\), there is exactly one \((i, j), i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i\) such that \(((i, j), k) \in Y\).

\(c_2\). Each copy is produced exactly once. That is, for each \((i, j), i = 1, 2, \ldots, n; j = 1, 2 \ldots, d_i\), there is exactly one \(k, k = 1, 2 \ldots, D\) such that \(((i, j), k) \in Y\).

\(c_3\). Lower indices copies are produced earlier. That is, if \(((i, j), k), (i, j'), k') \in Y\), and \(k < k'\), then \(j < j'\).

**Theorem 4.13** [74] For any feasible \(Y \subseteq \mathcal{Y}\),

\[
F_{\text{sum}} = \sum_{(i,j),k \in Y} C_{jk}^i + \sum_{i=1}^{n} \sum_{k=1}^{D} \inf f_i f_i(j - kr_i). 
\]

Note that an optimal solution cannot be obtained by simply solving the assignment problem only with the constraints \(c_1\) and \(c_2\). Theorem 4.11 holds true if \((i, j)\) is tied up with its ideal position \(\lceil \frac{2j-1}{2r_i} \rceil\). Constraint \(c_3\) ties up \((i, j)\) to its ideal position, and thus the result becomes an inequality without \(c_3\).

There is an optimal solution for \(F_{\text{sum}}^A\) which is feasible for \(F_{\text{sum}}\). The following feasibility theorem solves this problem in \(O(D)\) steps.

**Theorem 4.14** [74] If \(Y\) satisfies the constraints \(c_1\) and \(c_2\), then \(\hat{Y}\) satisfying \(c_1, c_2\) and \(c_3\) with \(c(\hat{Y}) \leq c(Y)\), \(c(Y) = \sum_{(i,j),k \in Y} C_{jk}^i\) can be determined in \(O(D)\) time. Moreover, each copy in the sequence \(s\) from \(\hat{Y}\) preserves the order that it has in the sequence \(s\) from \(Y\).
Each copy in the sequence \( s \) from \( \hat{Y} \) preserves order in the sequence \( s \) from \( Y \) since the assignment cost holds Monge inequality i.e., \( C_{jk}^i + C_{(j+1)k}^i \leq C_{jk}^{i} + C_{(j+1)k}^{i} \) with \( k < \hat{k} \) [73].

The following theorem provides the construction of the optimal sequence [74].

**Theorem 4.15** If \( \{x_{jk}^i\}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i; k = 1, 2, \ldots, D \) be an optimal solution for the assignment problem \( F_{sum}^A \) that preserves order, then
\[
\sum_{l=1}^{k} \sum_{j=1}^{d_i} x_{jl}^i, \ i = 1, 2, \ldots, n; \ k = 1, 2, \ldots, D, \]
is an optimal solution to the problem \( F_{sum} \).

### 4.5.2 Cyclic Sequences

The optimal cyclic JIT sequences exist for the problem. The existence of cyclic sequence as a solution to the total PRVP \( (F_{sum}) \) is indicated in [85, 86] and the optimality of the JIT cyclic sequences is shown in [67]. The cyclic sequencing is a factoring idea having potentiality to reduce the computational effort in constructing optimal sequences.

The existence of a cyclic sequence means the existence of a concatenation \( sss \ldots, m \) times, denoted by \( s^m \), \( m \in Z^+ \), of a sequence \( s \) of an instance \( (d_1, d_2, \ldots, d_n) \). The instance \( (md_1, md_2, \ldots, md_n) \) is considered here for the concatenation \( s^m \). The existence of an optimal cyclic sequence is the existence of an optimal \( s^m \) of optimal \( s \) [67]. This type of sequence can be found under the assumption \( F_i = F, \ i = 1, 2, \ldots, n, \) where \( F \) is convex and symmetric with minimum 0 at 0 [16]. The result is extended to be true if all \( F_i \) are convex, symmetric and equal in the interval \( (0, 1) \), but not true even if a single \( F_i \) is asymmetric [66].

The optimality of cyclic sequences depend on the two observations: the first is the nature of concatenation for Problem \( F_{sum} \). If \( z = uv \), where \( u \) and
v are sequences for the instances \((αd_1, αd_2, \ldots, αd_n)\) and \((βd_1, βd_2, \ldots, βd_n)\) with positive integers \(α\) and \(β\), then \(F_{\text{sum}}(z) = F_{\text{sum}}(u) + F_{\text{sum}}(v)\) [85]. The second observation is that the total cumulative production of model \(i, i = 1, 2, \ldots, n\), for the sequence \(z\) and that for any optimal sequence \(z^*\) exactly meets the demands. That is, if \(x(z)_{iD} = d_i, \ i = 1, 2, \ldots, n\), then there still exists an optimal sequence \(z^*\) such that \(x(z^*)_{iD} = d_i\) for all \(i\) [16]. This is based on swapping of two copies of different models so that the cost of the sequence does not increase. Generally, swapping procedure may increase the cost for different \(F_i, i = 1, 2, \ldots, n\), however there exists a special swapping procedure that does not increase the cost [67]. This method is limited to even instances that means instances with all demands being even. Furthermore, in this swapping process, the copies of models occupy positions at the same distance from the ends of the sequence.

The assumption of even instance ensures the symmetry on the costs. However, note that the symmetry on \(F_i\)'s does not mean the symmetry on the costs. This is because the ideal vertices \(\frac{2j-1}{2r_i}\) do not fall at the middle of the two integers \(Z_j^* - 1\) and \(Z_j^*\). The symmetry on the costs means the cost for \(j\) followed from 1 to \(D\) is equal to the cost for \(d_i + 1 - j\) followed from in the opposite direction from \(D\) to 1, [67, 73]. This means we have \(C^i_{(d_i+1-j)(D+1-k)} = C^i_{j+1-k}\) and \(C^i_{(d_i+1-j)k} = C^i_{j(D+1-k)}\) for any \(i = 1, 2, \ldots, n; k = 1, 2, \ldots, D\) and \(j = 1, 2, \ldots, d_i\).

The even instances of the form \(2d_1, 2d_2, \ldots, 2d_n\) for some positive integers \(d_1, d_2, \ldots, d_n\) are considered with feasible sequences of length \(2D\) such that \(\sum_{i=1}^{n} d_i = D\). The process of constructing a feasible sequence \(u\) for \(d_1, d_2, \ldots, d_n\) uses a specific function [67] that consists of three steps: folding, shuffle and unfolding, \(FSU\) in short. These operations are used to prove the existence of optimal cyclic sequences rather than to determine optimal sequences. The folding operation folds the sequence \(z = s_1, s_2, \ldots, s_D, s_D+1, \ldots, s_{2D}\) for \(2d_1, 2d_2, \ldots, 2d_n\) into the sequence \((s_1, s_{2D}), \ldots, (s_k, s_{2D+1-k}), \ldots, (s_D, s_{D+1})\)
of $D$ ordered pairs. The shuffle operation shuffles each pair producing a sequence $(z_1, z_{2D}), \ldots, (z_k, z_{2D+1-k}), \ldots, (z_D, z_{D+1})$ where 
\[
\{z_k, z_{2D+1-k}\} = \{z_k, z_{2D+1-k}\} \text{ for } k = 1, 2, \ldots, D.
\]
It uses a special case of the Hall’s theorem for the existence of a complete matching in a regular bipartite graph $G = (V_1 \cap V_2, E)$ where $V_1 = \{(i, j): i = 1, 2, \ldots, n; j = 1, 2, \ldots, \frac{d_i}{2}\}$, $V_2 = \{k: k = 1, 2, \ldots, \frac{D}{2}\}$ and $E = C(s)$, the edge set with nodes having degree 1 or 2 and $C(s)$ being the cost for the sequence $s$. The unfolding operation unfolds the shuffled sequence of pair into the feasible sequence $z = z_1, z_2, \ldots, z_D, z_{D+1}, \ldots, z_{2D}$ for $2d_1, 2d_2, \ldots, 2d_n$ [67].

The optimal sequence for the original problem can be obtained in the three steps (i) by calculating the greatest common divisor $m$ of $d_1, d_2, \ldots, d_n$, (ii) by using the algorithm in [74] to obtain an optimal sequence for $\frac{d_1}{m}, \frac{d_2}{m}, \ldots, \frac{d_n}{m}$, and (iii) by concatenating the sequence $m$ times to construct an optimal sequence for the original demands $d_1, d_2, \ldots, d_n$. The following two theorems provide the existence and the optimality of the cyclic sequences for total PRVP respectively [67].

**Theorem 4.16 (Existence theorem of cyclic sequence):**

If $z = s_1, s_2, \ldots, s_D, s_{D+1}, \ldots, s_{2D}$ is a feasible sequence for $2d_1, 2d_2, \ldots, 2d_n$, then a sequence $z' = z_1, z_2, \ldots, z_D, z_{D+1}, \ldots, z_{2D}$, where $i$ occurs exactly $d_i$ times in each of the two halves $z_1, z_2, \ldots, z_D$ and $z_{D+1}, \ldots, z_{2D}$, can be constructed such that $Z_{\min}(z') \leq Z_{\min}(z)$. 

81
Theorem 4.17 (Optimality theorem of cyclic sequence): 
If $\beta$ is an optimal sequence for the total PRVP (3.2) with the integer demands $d_1, d_2, \ldots, d_n$, then $\beta^m (m \geq 1)$ is optimal sequence for the problem with the demands $md_1, md_2 \ldots, md_n$.

Apart from reducing computational time complexity, the iterative nature of JIT cyclic sequences provides an important theoretical support to the JIT systems for repeating relatively short sequence to build a sequence for a longer time horizon. The time complexity depends on the magnitude of demands $d_1, d_2, \ldots, d_n$ and hence on $D$. Based on $D$ and $n$, the only known polynomial time optimization algorithm for JIT sequences has the time complexity $O(D^3)$ [67, 74, 76].

4.5.3 Weighted Bipartite Matching

The total PRVP (i.e., $F_{\text{sum}}$) can be reduced to a weighted matching problem in a complete bipartite convex graph, since $(i, j), i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i$, can be produced at any time unit $k, k = 1, 2, \ldots, D$. The assignment cost $C_{jk}^i$ for $(i, j)$ when sequenced at $k$ is taken as the weight for the edge $(i, j), k)$. The complete graph requires the calculation of $D^2$ weights [104]. The number of calculated weights can be substantially reduced with an appropriate bottleneck $B \leq 1$ from the complete convex bipartite graph calculating only those weights for which $k \in [E(i, j), L(i, j)]$, where $E(i, j)$ and $L(i, j)$ are the earliest and the latest sequencing times respectively (See Section 4.2 for convex bipartite graph and $E(i, j)$ and $L(i, j)$). With this, the complete convex bipartite graph is transformed into an incomplete graph $G(B \leq 1)$.

An optimal sequence always exists in the complete convex bipartite graph whereas it is not guaranteed in the incomplete convex bipartite case. A computational analysis shows that optimal solutions to the min-sum problem
with absolute-deviation or square-deviation objective in $G(B \leq 1)$ also optimal in the complete graph [66]. This result can be generalised for identical, symmetric and convex cost functions over the range $[-1, 1]$. This feature is known as oneness property [27]. The theorem 4.20 best illustrates the above discussion [66].

**Theorem 4.18** An optimal solution to the total PRVP ($F_{sum}$) with $f_i = f$, $i = 1, 2, \ldots, n$, and $-1 \leq x_{ik} - kr_i \leq 1$ for the incomplete convex bipartite graph is also optimal solution to the $F_{sum}$ problem for the complete bipartite convex graph.

As an immediate result, the optimal solutions to $F_{sum}$ problem with identical, symmetric and convex cost functions can be obtained by reducing the problem into weighted bipartite matching problem constructed on $G(B \leq 1)$. With this reduction, significant amount of solution time and storage space can be obtained. A further reduction can be obtained for instances with the greatest common divisor of the demands $d_1, d_2, \ldots, d_n$ greater than 1 [15, 66].

Note that the result cannot be generalised for non-identical cost functions in the range $[-1, 1]$ such that $-1 \leq x_{ik} - kr_i \leq 1$. That is, the cost functions can be shown to exist in which the optimal min-sum sequence always contains cases where actual production for some product deviates from its desired level of production by more than one unit. The following counter example in [66, 60] better illustrates the situation:

Consider 10 products whose demand vector is

$$(16, 24, 24, 28, 28, 42, 42, 42, 42, 48)$$

and the respective cost functions are $f_1(z) = f_2(z) = \alpha_1 \cdot |z|$, $f_3(z) = f_4(z) = \alpha_2 \cdot |z|$, $f_5(z) = f_6(z) = f_7(z) = f_8(z) = \alpha_3 \cdot |z|$, $f_9(z) = \alpha_4 \cdot |z|$, $f_{10}(z) = \alpha_5 \cdot |z|$, where $z = x_{ik} - kr_i$ and $\alpha_1 = 168^2 \times 8 \times 7 \times \alpha_2$, $\alpha_2 = 168^2 \times 6 \times 6 \times \alpha_3$, $\alpha_3 = 168^2 \times 4 \times 2 \times \alpha_4$, $\alpha_4 = 168^2 \times 7 \times \alpha_5$, $\alpha_5 = 1$. 
This concludes that for any optimal sequence $s$ of the above instance, there exist some product $i'$ and time $k$ such that $|x(s)_{i'k} - kr_{i'}| > 1$. Hence the solutions to the min-sum problems with non-identical cost functions need not satisfy the oneness property and can not be found solely by solving the problem in $\mathcal{G}(1)$ [66].

Weight function that gives rise to an optimal solution to the $F_{\text{sum}}$ problem and vice versa can be found in [79] as the following.

**Theorem 4.19** A sequence $s$ is optimal to the $F_{\text{sum}}$ problem if and only if there is a minimum weight perfect matching $\mathcal{M}$ with a weight function $w : V_1 \cup V_2 \to R$ such that

$$w_k + w_{ij} \leq C^i_{jk} \text{ for all edges } ((i, j), k)$$

and $\sum_{k \in V_1} w_k + \sum_{(i,j) \in V_2} w_{ij} = \sum_{((i,j), k) \in \mathcal{M}} C^i_{jk} = s$.

This theorem shows a compact way for an optimal solution to the $F_{\text{sum}}$ problem by providing the the corresponding perfect matching and the vertex weight values $w_{ij}$ in the graph $\mathcal{G}_w$.

In the convex bipartite graph $\mathcal{G}(B \leq 1)$, the number of edges is $|\mathcal{E}| \leq nD + 2D$ [104]. This shows that the calculation of the edge weights requires $O(|\mathcal{E}|) = O(nD)$ time. The assignment algorithm for the weighted convex bipartite graph requires $O(|V||\mathcal{E}| \log |V|)$ time, where $V = V_1 \cup V_2$ and $|V| = 2D$ [96]. Hence, the minimum weight perfect matching in the weighted convex bipartite graph $\mathcal{G}(B \leq 1)$ can be determined in $O(nD^2 \log D)$ time.

This type of optimal solution hardly exists for the instances of practical size. The question of determining minimum $B$ such that the optimal solution to the $F_{\text{sum}}$ problem is $B$-bounded remains unanswered [33]. Recall that a solution is said to be a $B$-bounded or a $B$-feasible if the deviation is less than a given bottleneck $B$. It is well-established that an upper bottleneck on the optimal value of the $F_{\text{sum}}$ problem with absolute-deviation objective
or square-deviation objective is $O(nD)$ though the bottleneck is not tight [104]. Nevertheless, a lower bottleneck to the problem $F_{\text{sum}}$ with the square-deviation objective is $\sum_{i=1}^{n} \frac{D^2 - d_i^2}{12D}$ [1].

Any sequence $s$ is said to have oneness property if and only if $s$ is obtained from $\mathcal{G}(B \leq 1)$, [66]. Let $S_1$ be the set of all feasible sequences with oneness property. Under this property, the cost matrices $C_{jk}^{i}$, $i = 1, 2, \ldots, n; j = 1, 2, \ldots, d_i; k = 1, 2, \ldots, D$, of a feasible sequence with $\mathcal{G}(B = 1)$ for absolute deviation min-sum objective $F_{\text{sum}}^{a'}$ and squared-deviation objective $F_{\text{sum}}^{s'}$ are equivalent. That is, the cost matrices for the objective functions (3.11) and (3.12) are same. If $C_{a'jk}^{i}$ and $C_{s'jk}^{i}$ are the cost matrices to problems $F_{\text{sum}}^{a'}$ and $F_{\text{sum}}^{s'}$ respectively, then we readily have the following lemma [79].

**Lemma 4.1** For $B = 1$, $k \in [E(i, j), L(i, j)]$ implies $C_{a'jk}^{i} = C_{s'jk}^{i}$, for all $i, j, k$.

The cost matrices for problem $F_{\text{sum}}^{a'}$ and problem $F_{\text{sum}}^{s'}$ may not be the same for $B < 1$. Consequently, both the problems may not have optimal sequences on $S_1$. Moreover, an optimal sequence to the problem $F_{\text{sum}}^{s'}$ may belong to $S_1$ but not an optimal sequence to problem $F_{\text{sum}}^{a'}$ belong to $S_1$ for the same instance. As a counter example, the instance $(46, 46, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ has no optimal sequence to problem $F_{\text{sum}}^{a'}$ in $S_1$ [27]. If problem $F_{\text{sum}}^{s'}$ has no optimal sequence in $S_1$, then problem $F_{\text{sum}}^{a'}$ has no optimal sequence. If it had, the sequence would be optimal to problem $F_{\text{sum}}^{s'}$ also. Although problem $F_{\text{sum}}^{a'}$ may not have optimal sequence in $S_1$, it provides a lower bottleneck and an upper bottleneck to problem $F_{\text{sum}}^{s'}$ [27]. It is important that an optimal sequence to problem $F_{\text{sum}}^{a'}$ in $S_1$ is also optimal to problem $F_{\text{sum}}^{s'}$. The following theorem clearly reveals this fact [27].

**Theorem 4.20** Any optimal solution to problem $F_{\text{sum}}^{a'}$ in $S_1$ is also optimal to problem $F_{\text{sum}}^{s'}$ but not conversely.
As a result, this concludes that problem $F_{sum}^{s'}$ can be solved by means of solving problem $F_{sum}^{a'}$ in $S_1$. This is important because the transformation of absolute penalties required is smaller in magnitude than that of the square penalties \[66\].

Two problems are said to be $S_1$-equivalent if both the problems have the same set of optimal sequences in $S_1$. The problems $F_{sum}^{a'}$ and $F_{sum}^{s'}$ are $S_1$-equivalent. However it is noteworthy that the optimal sequences to the problem $F_{sum}^{a'}$ and the problem $F_{sum}^{s'}$ without having oneness property may not coincide \[27\].

The $S_1$-equivalence between these two problems suggests a conjecture about the problems $F_{sum}$ to be $S_1$-equivalent. A computational result in \[66\] indicate that the $S_1$-equivalence between problem $F_{sum}^{a'}$ and problem $F_{sum}^{s'}$ may be due to symmetry and convexity of their objective functions. However, this conjecture is not true. A counter example \[27\] is the instance of $n = 6$, $D = 50$ such that $(23, 23, 1, 1, 1, 1)$ with the convex objective function defined by

$$f(x_{ik} - kr_i) = \begin{cases} \frac{-x_{ik} - kr_i}{1-B^*} - \frac{B^*}{1-B^*}, & (x_{ik} - kr_i) \leq -B^* \\ 0, & -B^* \leq (x_{ik} - kr_i) \leq B^* \\ \frac{x_{ik} - kr_i}{1-B^*} - \frac{B^*}{1-B^*}, & B^* \leq (x_{ik} - kr_i) \end{cases},$$

where $B^*$ is the optimal value of absolute deviation bottleneck PRVP, $F_{max}^a$.

Thus, in conclusion the absolute deviation objective $F_{sum}^{a'}$ and squared deviation objective $F_{sum}^{s'}$ have the same set of optimal solutions under the oneness properety; however this property can not be generalized to other convex, nonnegative and symmetric functions. Both the problems having optimal solutions with the oneness property indicate that solving one of the problems is equivalent to solve the other one.
4.6 Sequencing over Bicriterion Objectives

In the previous Sections, we observed that both the bottleneck PRVP and the total PRVP are pseudo-polynomially solvable. In this Section, we observe bicriteria sequencing objectives that a sequence can be optimal with respect to both the objectives. This bicriterion approach combines the desirable properties of both criteria. It is shown that several Pareto optimal solutions may be efficiently determined by using a Pareto algorithm [104]. A perfect matching for the convex bipartite graph corresponding to the bound $B$ is obtained in the Pareto algorithm. Then an order preserving perfect matching with a minimum weight is determined.

If there is a perfect matching in a graph $G(B)$, then a minimum weighted perfect matching also exists. Moreover, we note that a perfect matching always exists in $G(1)$. The assignment method for total PRVP to find a minimum weight perfect matching in bipartite graph $G = (V, E)$ can be implemented in $O(|V||E| log|V|)$ time[96]. Thus, the time complexity of bicriterion sequencing depends on the number of edges $|E|$ in $G(B)$.

Since $|E|$ is computed by the value of $B$, the time complexity of the bicriterion algorithm must also depend upon $B$. For $B \leq 1$, it is shown that $|E|$ is significantly smaller than the edge set of a complete graph, thereby allowing for a reduction in the time complexity of the bicriterion problem in comparison to that of the min-sum procedure. The solutions of these bicriterion problems also deserves many highly desirable properties that significantly add to the efficiency of the procedure [104]. We restate that the upper and lower bounds for bottleneck PRVP are 1 and $1 - r_{max}$ where $r_{max} = \max_i \{r_i\}$. The upper bound for the total PRVP with absolute and square deviation is $nD$.

**Lemma 4.2** [104] The number of edges generated in the graph $G(B)$ with $B \leq 1$ is $|E| \leq (n + 2)D$. 

87
As an instance, for $n = 10$ and $D = 1,000$ with $B \leq 1$, there will be only 12,000 penalties to be calculated for the weighted matching problem in $G(B)$, which is substantially fewer than the 1,000,000 calculations required in the complete assignment formulation for $D^2$ penalties. Therefore, in-core memory requirements for this instance of the problem are significantly reduced.

**Lemma 4.3** [104] The assignment solution for total PRVP in the convex bipartite graph $G(B)$ with $B \leq 1$, can be determined in $O(nD^2\log D)$.

**Proof:** The number of edges in the graph $G(B)$ is $|E| \leq nD + 2D$ by lemma 4.2. The time required to calculate the weights of edges is $O(|E|) = O(nD)$. The assignment solution for total PRVP in weighted bipartite graphs requires $O(|V||E|\log|V|)$ time. Noting $|V| = 2D$ and on substitution of the above values, the optimal assignment can be determined in $O(2nD^2\log 2D)$ time, and hence in $O(nD^2\log D)$ time.

If $S$ and $\beta$ are feasible solutions of total PRVP and bottleneck PRVP, then the bicriterion JIT level sequence is determined by the following lemma:

**Lemma 4.4** [104] A bicriterion just-in-time level schedule with a solution $(S(1), \beta \leq B = 1)$ can be found in $O(nD^2\log D)$.

If $S^*$ and $B^*$ are the optimal values for the total PRVP and bottleneck PRVP respectively, then it can be observed that $S^* \leq S(1)$ and $B^* \leq B = 1$. Therefore, the sequence may not be optimal with respect to either measure. With some adjustments to the method of Lemma 4.4, a Pareto optimal solution can easily be determined using a similar efficient approach.

**Theorem 4.21** [104] The Pareto optimal solution $(S(B^*), B^*)$ can be determined in $O(nD^2\log D)$ time.
This solution procedure runs faster than $O(D^3)$ in general, since in practice, normally we have the case that $n \leq D$. Thus, the time complexity for determining a Pareto optimal solution is lower than that for determining an optimal min-sum solution alone. That is, \((i.e., O(nD^2\log D)) \leq O(D^3)\) in most of the cases.

Furthermore, with only minor modifications to this approach, all of the Pareto optimal solutions with $\beta \leq 1$ can be efficiently generated. The following algorithm can be used to determine all such Pareto optimal solutions.

**Algorithm: Pareto** [104]

1. Initialize $B \leftarrow 1 - r_{max}$.

2. Generate the edge set for the convex bipartite graph corresponding to $B$ and denote this edge set by $E(B)$.

3. Determine a perfect matching, if it exists, in this bipartite graph.
   If there is no perfect matching, then let $B \leftarrow B + \frac{1}{D}$, and go to step 2.
   If a perfect matching exists, then determine a minimum weight order preserving perfect matching in $G(B)$. The corresponding production sequence will be Pareto optimal with min-max value $B = B^*$ and the min-sum value $S = S(B^*)$.
   Let $E = E(B)$, $S_{\text{min}} = S(B^*)$ and go to step 4.

4. Let $B \leftarrow B + \frac{1}{D}$. If $B > 1$, then stop - all required Pareto optimal solutions with $B \leq 1$ have been determined. Otherwise, go to step 5.

5. Generate the edge set $E(B)$ for the convex bipartite graph corresponding to $B$.
   If $E(B) = E$, then go to step 4.
   Otherwise set $E = E(B)$ and go to step 6.

6. Determine a minimum weight order preserving perfect matching in $G(B)$.  

89
If $S(B) < S_{\text{min}}$, then this sequence will be Pareto optimal with $\beta = B$ and $S = S(B)$, set $S_{\text{min}} = S(B)$ and go to step 4.

Otherwise the sequence is not Pareto optimal. Go to step 4.

Algorithm Pareto determines the production sequences for all Pareto optimal solutions $(S, \beta)$, with $\beta < 1$ in $O(nd_{\text{max}}D^2\log D)$ time [104], where $d_{\text{max}} = \max\{d_1, d_2, \ldots, d_n\}$.

The algorithm Pareto implies that many Pareto optimal solutions can be efficiently found in $O(nd_{\text{max}}D^2\log D) < O(nD^3\log D)$ time. Determining all of these production sequences requires the addition of a factor of less than $n\log D$ to the time complexity of the min-sum assignment procedure in [74, 71]. The upper bounds have been shown for two commonly used min-sum objectives; the sum of squared and sum of absolute deviations. Since $\beta \leq 1$ in the Pareto algorithm, $S(\beta) \leq nD$ for either of these two formulations. Hence, the upper bounds are readily established on the objective function values of the bicriterion solution using either of these common min-sum objectives [104].
Chapter 5

Discrete Apportionment Approach: An Efficient Frontier

Chapter 5 is based on the papers C, E, G, P.

The apportionment approach of discrete nature has been seen as one of the directions to tackle the JITSP in single-level case. The divisor methods, particularly its parametric cases, are shown efficient to obtain the comparative solutions of the JITSP. In this chapter, we present the apportionment solution to the sum deviation JITSP in the single-level.

We formulate the discrete apportionment problem in Section 5.1. The JITSP and the DAP are mutually transformed into each other in Section 5.2. A bird’s eye-view on apportionment methods is given in Section 5.3. The divisor methods are explained in Section 5.4 together with parametric divisor methods. In Section 5.5, we propose mean-based divisor methods presenting the efficient frontiers in Section 5.6 with respect to local and global deviations. In Section 5.7, we present the results of CA election 2008 of Nepal in PR system, providing slight changes in the existing allocations.
5.1 Formulation of Discrete Apportionment Problem

The discrete apportionment problem (DAP), as discussed in Chapter 2, has crucial role in maintaining social and political order of every nation. Here we formulate the DAP mathematically as follows:

Assume that there are \( s \) states (or parties) indexed \( i = 1, 2, \ldots, s \) in any nation, which are supposed to have representatives (or seats) according to the size of their population (or votes) from the congressional integer house size \( h \).

Suppose that the state \( i \) has a population \( p_i \) such that the total population of the nation is \( \sum_{i=1}^{s} p_i = p \). The fundamental problem is to apportion \( a_{ih} \) integer seats to state \( i \) under the constraints \( \sum_{i=1}^{s} a_{ih} = h \) and \( a_{ih} \in \mathbb{Z}^+ \). An ideal apportionment is assumed to satisfy the equation \( \frac{p_i}{p} = \frac{a_{ih}}{h} \) for all states, yielding \( a_{ih} = \frac{p_i h}{p} \), called ideal quota or fair share for state \( i \) denoted by \( q_{ih} \), and is not necessarily an integer. Hereafter, the actual and the ideal apportionments are denoted by \( a_{ih} \) and \( q_{ih} \), associated with the house size \( h \) respectively. Since only the integral \( a_{ih} \) can be assigned to any state, the crucial point is how to handle this problem fairly. One immediate idea is *rounding*: for each state, ideal apportionment should either be rounded down to the next lower integer or rounded up to the next higher integer; but should never exceed these bounds \([7]\). The lower and the upper bounds are defined by the floor and ceiling values of the fractional number \( q_{ih} \) as follows: \( [q_{ih}] \leq q_{ih} \leq \lceil q_{ih} \rceil \). The apportionment vector \( a \) satisfies the quota if and only if \( [q_{ih}] \leq a_{ih} \leq \lceil q_{ih} \rceil \) for each state \( i \). This idea of rounding is not unique. For fixed house size \( h \), the sum deviation global index of apportionment to be minimized is formulated as

\[
\text{minimize} \quad \sum_{i=1}^{s} (a_{ih} - q_{ih})^2 
\]  

(5.1)
5.1 Formulation of Discrete Apportionment Problem

The discrete apportionment problem (DAP), as discussed in Chapter 2, has a crucial role in maintaining social and political order of every nation. Here we formulate the DAP mathematically as follows:

Assume that there are $s$ states (or parties) indexed $i = 1, 2, ..., s$ in any nation, which are supposed to have representatives (or seats) according to the size of their population (or votes) from the congressional integer house size $h$.

Suppose that the state $i$ has a population $p_i$ such that the total population of the nation is $\sum_{i=1}^{s} p_i = p$. The fundamental problem is to apportion an integer seats to state $i$ under the constraints $\sum_{i=1}^{s} a_{ih} = h$ and $a_{ih} \geq 1$.

An ideal apportionment is assumed to satisfy the equation $p_i p = a_{ih} h$ for all states, yielding $a_{ih} = \frac{p_i h}{p}$, called ideal quota or fair share for state $i$ denoted by $q_{ih}$. Hereafter, the actual and the ideal apportionments are denoted by $a_{ih}$ and $q_{ih}$, associated with the house size $h$ respectively. Since only the integral $a_{ih}$ can be assigned to any state, the crucial point is how to handle this problem fairly. One immediate idea is rounding: for each state, ideal apportionment should either be rounded down to the next lower integer or rounded up to the next higher integer; but should never exceed these bounds.

The lower and the upper bounds are defined by the floor and ceiling values of the fractional number $q_{ih}$ as follows:

$$\lfloor q_{ih} \rfloor \leq q_{ih} \leq \lceil q_{ih} \rceil.$$  

The apportionment vector $a_i$ satisfies the quota if and only if $\lfloor q_{ih} \rfloor \leq a_{ih} \leq \lceil q_{ih} \rceil$ for each state $i$. This idea of rounding is not unique. For fixed house size $h$, the sum deviation global index of apportionment to be minimized is formulated as

$$\min \sum_{i=1}^{s} (a_{ih} - q_{ih})^2$$

such that $\sum_{i=1}^{s} a_{ih} = h$ and $a_{ih} \geq 1$.

which is a constrained integer programming problem seeking for integer allocations $a_{ih}$, in such a way that the sum of them does not exceed the house size $h$ and they remain near to the fair shares $q_{ih}$ as close as possible. Also the allocations must never be less than unity, since it is the minimum requirement.

The absolute deviation objective of the discrete apportionment problem can similarly be formulated as $\sum_{i=1}^{s} |a_{ih} - q_{ih}|$ under the same constraints.

5.2 JIT Sequencing versus Apportionment

Establishing the relation between PRVP and DAP, Bautista et al. [13] have shown that the JIT sequencing problem can be seen as a constrained sequential apportionment problem. The monotone condition of JIT sequencing problem is equivalent to house monotonicity in DAP. They indicated that the algorithm of Inman and Bulfin [56] to minimize the objective function (3.13) is the Webster divisor method of apportionment.

Balinski and Shahidi [5] proposed a strong approach to JIT sequencing via axiomatics, which are originally developed for the apportionment problem. The axiomatic method of apportionment depends on some socially desirable characteristics, such as satisfying quota, house monotonicity and population monotonicity, which must be satisfied for the solution of apportionment problem. It is difficult to find a perfect method that meets all the constraints. To this end, the impossibility theorem of Balinski and Young in [6] states that there is no perfect apportionment method satisfying all properties at a time.

The JIT sequencing problem in terms of parametric divisor methods is studied in [4]. Józefowska et al. [58] characterized some of the algorithms of
JIT sequencing via apportionment theory with suitable transformation of the problems. Adding some similar properties, we present the notational interrelation of the two problems in Table 5.1:

<table>
<thead>
<tr>
<th>JIT Sequencing verses Apportionment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of products $n$ $\iff$ number of states $s$</td>
</tr>
<tr>
<td>Product $i$ $\iff$ state $i$</td>
</tr>
<tr>
<td>Vector of demands $d$ $\iff$ vector of populations $p$</td>
</tr>
<tr>
<td>Demand $d_i$ for product $i$ $\iff$ population $p_i$ of state $i$</td>
</tr>
<tr>
<td>Position in sequence $k$ $\iff$ size of house $h$</td>
</tr>
<tr>
<td>Actual production $x_{ik}$ $\iff$ actual apportionment $a_{ih}$</td>
</tr>
<tr>
<td>Ideal production $kr_i$ $\iff$ exact quota $q_{ih}$</td>
</tr>
<tr>
<td>Demand $D = \sum_{i=1}^{n} d_i$ $\iff$ population $p = \sum_{i=1}^{s} p_i$</td>
</tr>
<tr>
<td>Monotone condition in JIT $\iff$ house monotone in DAP</td>
</tr>
</tbody>
</table>

Thus, the two problems can be observed from the same window and handled in similar ways in most cases, such as the parametric divisor methods of apportionment generate cyclic JIT sequences. Balinski and Shahidi [5] proposed other types of deviations for the pair of products targeting to minimize the variation of production rates from product to product, which we say is equivalent to the state to state variation of apportionment. In the following sections, we present apportionment methods, further characterizations and joint approaches for the two problems via global and local deviations.

### 5.3 On the Methods of Apportionment

The apportionment methods are vital elements of every social and political order which affect not only how collective decisions are made by respective authorities, but also how and why a particular constitutional order develops.
over time. As the outcome of many debates around middle of 1700s, there are several methods of apportionment in the literature of congressional apportionment theory proposed by various mathematical and political scientists in different time intervals and political situations. Some of them are tested and applied in real situations and some of them are discarded due to their impracticability. Confusingly, many of these methods have different names and facades, while being equivalent from a mathematical perspective. Some apportionment methods do not always yield unique solutions; for instance, when two parties receive the same number of votes. In this situation, a suitable tie-breaking rule is required.

From the analysis of apportionment history and existing literature, it can be observed that there are roughly three types of apportionment methods in general: (i) Hamilton-type methods, (ii) divisor methods, and (iii) a series of modern methods that combine elements from various techniques (e.g., power indices) and fields (e.g., social choice theory). Our main concern in this dissertation is with divisor methods. Here we do not touch the third category. However, keeping a bird’s eye-view in Hamilton-type of apportionment methods, we discuss on divisor methods in Section 5.4.

The class of Hamilton-type methods consists of two apportionment methods, namely Hamilton method and Lowndes method. Their intuitive and simple algorithm is their main appeal and as a result, its basic algorithm is still widely in use. However, it appeared that these methods are heavily defective due to Alabama paradox.

In Hamilton method, the apportionments are made easily as follows: compute the quota and assign to each state its integer part. Distribute unapportioned seats to the states ordering with the largest remainders until the house is full. This method was used in the U.S. to distribute the seats in the house of representatives from 1850 to 1900, and so it is important from a historical perspective. This method, proposed by A. Hamilton in 1792, seems to be
natural and simple considering the quota approach. The following algorithm shows a formal description of how the apportionment process is carried out in Hamilton method.

**Algorithm Hamilton** (Largest Reminder Method)

**Step 1:** Give each constituency as many seats as the integer part of its quota \([q_i]\).

**Step 2:** Order the fractional parts \(r_i\) in descending order. If there is a tie for a position, break it in favour of any of the eligible fractions. Find the sum of all fractional parts \(r_m\); either as \(r_M = h - \sum_{i \in M} [q_i]\) or \(r_M = \sum_{i \in M} r_i\). Give one seat each to the first \(r_M\) constituencies on the ordered fraction list.

**Step 3:** The final apportionment to a constituency is the sum of the seats given to it in step 1 and 2.

The Lowndes’ method has been similarly defined in literature. Later, the number of citizens associated to one seat was first considered which corresponds to the divisor methods. Referring [6, 112] for some details on several methods of apportionment, we provide a brief description of divisor methods in Section 5.4, and hence we propose mean-based divisor methods in Section 5.5 developing the efficient frontier in Section 5.6.

### 5.4 Divisor Methods

The divisor methods comprise a family of monotone methods involving a notion of rounding, each of which is defined by a monotone increasing divisor function \(d(a)\) such that \(a \leq d(a) \leq a + 1\). The divisor methods are also known as the methods of highest averages, which vary according to the form of divisors. Huntington [55] made a systematic study of divisor methods (also called Huntington methods) based upon the rank-index \(r(p, a) = \frac{p}{d(a)}, d(a) \neq 0\) and the fairness measure \(\frac{a_{ih}}{p_i} > \frac{a_{jh}}{p_j}\), minimizing pairwise measure of inequity between the states \(i\) and \(j, i = 1, 2, \ldots, n; j = 1, 2, \ldots, n\), where \(p\) and \(a\)
represent the population and the apportionment vectors respectively [7, 8]. The state achieving maximum of $\frac{p_i}{d(a)}$ gains the $(h + 1)^{th}$ seat. Note that for given house size $h$, $\frac{p_i}{a_i}$ and $\frac{a_i}{p_i}$ represent the average district sizes and the share of representatives respectively of state $i$. Also note that an apportionment method $M$ is said to be house monotone if for every apportionment solution $f \in M$, we have $f(p, h) \leq f(p, h + 1)$.

Practically, there always exists a certain inequality between two states, which gives one of the states a slight advantage over the other. The state $i$ is better off than state $j$, if $\frac{a_{ih}}{p_i} > \frac{a_{jh}}{p_j}$. An apportionment $a$ is stable if no transfer of one seat from a better off state $i$ to a less well off state $j$ reduces the value of the inequality. The divisor functions and fairness measures of inequity in the divisor method are not unique [111, 112]. That is, the inequality $\frac{a_{ih}}{p_i} > \frac{a_{jh}}{p_j}$ can be rearranged by cross-multiplication in $2^4 = 16$ different ways by taking different combinations of $p_i, p_j, a_i, a_j$ [6]. In fact, Huntington [55] examined 64 different measures of local inequity considering different combinations of $p_i, p_j, a_i, a_j$ between two states including 32 relative and 32 absolute differences. All of the relative differences and two of the absolute differences lead to MEP. His noble approach is pairwise transfer of seats among states according to the priority basis to balance the apportionment. A transfer is made from the more favored state to less favored state if this reduces the inequity measure between two states.

There are five divisor methods basically known as Huntington methods. Without loss of generality, we use $A$ for Adams (smallest divisor) method, $D$ for Dean (harmonic mean) method, $H$ for Hill (equal proportion) method, $W$ for Webster (major fraction) method and $J$ for Jefferson (largest divisor) method. The methods $A$ and $D$ were never in use, the method $H$ is in practice since 1940 to date, the method $W$ was used in 1840, 1910 and 1930, and the method $J$ was used during the time interval 1792-1830. Keeping these notations in mind, we present the details of the five traditional divisor methods in Table 5.2.
Table 5.2: The five divisor methods of apportionment

<table>
<thead>
<tr>
<th>Method</th>
<th>Mathematical name</th>
<th>Divisor $d(a)$</th>
<th>Rank-index $r(p, a)$</th>
<th>Fairness $a_i/p_i &gt; a_j/p_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Smallest divisor</td>
<td>$a$</td>
<td>$\frac{p}{a}$</td>
<td>$a_i - a_j (\frac{p_i}{p_j})$</td>
</tr>
<tr>
<td>$D$</td>
<td>Equal proportion</td>
<td>$\sqrt{a(a+1)}$</td>
<td>$\frac{p}{\sqrt{a(a+1)}}$</td>
<td>$\frac{p_i}{a_j} - \frac{p_i}{a_i}$</td>
</tr>
<tr>
<td>$H$</td>
<td>Harmonic mean</td>
<td>$\frac{2a(a+1)}{2a+1}$</td>
<td>$\frac{p}{2a(a+1)}$</td>
<td>$\frac{a_ip_j}{a_jp_i} - 1$</td>
</tr>
<tr>
<td>$W$</td>
<td>Major fraction</td>
<td>$a + \frac{1}{2}$</td>
<td>$\frac{p}{a+\frac{1}{2}}$</td>
<td>$\frac{a_i}{p_i} - \frac{a_j}{p_j}$</td>
</tr>
<tr>
<td>$J$</td>
<td>Largest divisor</td>
<td>$a + 1$</td>
<td>$\frac{p}{a+1}$</td>
<td>$a_i (\frac{p_i}{p_j}) - a_j$</td>
</tr>
</tbody>
</table>

If a tie occurs between states with unequal populations (however extremely unlikely), Huntington suggested that it should be broken in favor of the larger state. His approach made remarkable use of pairwise comparison of local measures of inequity to be minimized between two states. Such pairwise comparisons often appear more appropriate in social sciences, if one is seeking to arrive at a fair outcome. Huntington-Hill method considers the method of equal proportion (MEP) with relative differences in both district sizes and shares of a representative as given below:

$$\left| \frac{p_i}{a_i} - \frac{p_j}{a_j} \right| / \min \left( \frac{p_i}{a_i}, \frac{p_j}{a_j} \right)$$

and

$$\frac{a_i}{p_i} - \frac{a_j}{p_j} / \min \left( \frac{a_i}{p_i}, \frac{a_j}{p_j} \right)$$

Huntington [55] showed that MEP is the best of the five divisor methods, since it relies on the most natural measure of inequality— the relative difference. His proposition was supported by two selection committees which reported to the president of the National Academy of Sciences in 1929 [18] and in 1948 [92]. Both of these reports pleaded for MEP because it is unambiguous and house monotone yielding apportionments that are neutral
with respect to emphasis on larger and smaller states. Moreover, MEP is consistent: If \((p, a)\) and \((p', a')\) are tied (i.e., \(p = p'\)), then any method \(M\) should be indifferent between such states. That is, for some \(p\) and \(h, f_i(p, h) = a, f_j(p', h) = a'\), if a solution \(f \in M\) gives the \((h + 1)^{st}\) seat to state \(i\), then there should be an alternative solution \(g \in M\) identical with \(f\) up to \(h\) (i.e., \(f_h = g_h\)) that gives the \((h + 1)^{st}\) seat to state \(j\). Moreover, consistency means if \((p, a) \sim (p', a')\), then any two states with populations \(p\) and \(p'\), and apportionments \(a\) and \(a'\) equally deserve an additional seat under the operation of the method \(M\). Huntington divisor method is both house monotone and consistent. Moreover, Balinski and Young proved the following theorem with two key lemmas in [8].

**Theorem 5.1** An apportionment method \(M\) is house monotone and consistent if and only if it is a Huntington method.

Note that some inequality measures are unworkable, which may lead to infinite cycling of solutions, e.g., \(\frac{a_i}{a_j} - \frac{p_i}{p_j}\) and \(\frac{p_j a_j}{p_i} - \frac{1}{a_i}\), where \(\frac{p_i}{a_i} = \frac{p_j}{a_j}\). Keeping deep insights to the traditional divisor methods, Oyama [95] gave ARPT (average ratio pairwise transfer) rule, which implied larger stable region than Huntington’s one. He viewed the apportionment methods from the angle of constrained optimization problem, restricting its application to the case when one state is over-represented absolutely and the other under-represented absolutely, that is, when \(\frac{a_i}{p_j} \geq \frac{\sum_i h_{p_i}}{s_{i p_i}} \geq \frac{a_j}{p_j}\) [9]. The main idea of divisor (rank-index) methods is:

The Rank-index method

Step 1. Start with \(f(p, 0) = 0\), i.e., \(a_i = 0\), \(i = 1, 2, \ldots, s\)

Step 2. Find a state \(t\) such that \(r(p_t, a_t) = \max_i r(p_i, a_i)\)

Then \(a'_t = a_t + 1\) and \(a'_j = a_j\) for \(j \neq t\)

Step 3. Repeat step 2 until all \(h\) seats are allocated.

We describe parametric divisor methods in the following Subsection.
5.4.1 Parametric Methods

Any divisor method \( \phi^d \) is called parametric method \( \phi^\delta \), if \( d(a) = a + \delta \) where \( 0 \leq \delta \leq 1 \). The methods \( A, W \) and \( J \) are parametric with \( \delta = 0, 0.5, 1 \) respectively. Saint-Lague favored \( W \) and d'Hondt favored \( J \). Condorcet proposed slightly different parameter \( \delta = 0.4 \), Thapa and Dhamala [112] proposed \( \delta = 0.7 \). Being linear divisor functions, parametric methods are computationally very efficient. Moreover, they are cyclic generating cyclic JIT sequences: for two instances of the JIT sequencing \( D_1 = d_1, d_2, \ldots, d_n \) and \( D_2 = kD_1 = kd_1, kd_2, \ldots, kd_n \), the sequence for \( D_2 \) is obtained by \( k \) repetitions of the sequence for problem \( D_1 \). As \( \delta \) increases from 0 to 1, seats are given-up by the smaller states in favor of the larger states [4]. That is, a parametric method \( \phi^\alpha \) gives-up to another parametric method \( \phi^\beta \) if and only if \( \alpha < \beta \). Thus, parametric method \( \phi^\delta \) is most favorable to smaller states with \( \delta = 0 \) and most favorable to larger states with \( \delta = 1 \).

**Lemma 5.1** A parametric method \( \phi^\alpha \) gives-up to another parametric method \( \phi^\beta \) if and only if \( \alpha < \beta \).

The three fundamental properties of parametric method are as follows:

**Anonymity:** the solutions must depend only on the values of the data, not on the order in which the data is presented. **Scale-invariance:** \( \phi(p, h) = \phi(\lambda p, h) \), for all \( \lambda > 0 \); **Exactness:** if \( p \) is integer valued and \( \sum_i p_i = h \) then \( p \) is the unique solution \( \phi(p, h) = p \). A method \( \phi \) is balanced if \( a \in \phi(p, h) \) and \( p_i = p_j \) implies \( |a_i - a_j| \leq 1 \).

**Lemma 5.2** A consistent, exact and anonymous method is balanced.

An apportionment method \( \phi \) is cyclic, if \( a \in \phi(p, h), p \) integer implies \( (a + p) \in \phi(p, h + p_s) \), for example Hamilton method is cyclic.
Moreover, we have the following equivalency of parametric and cyclic natures.

**Theorem 5.2** A divisor method $\phi$ is parametric if and only if it is cyclic.

There are infinitely many parametric divisor functions lying between $a$ and $a + 1$ depending upon the value of $\delta$ such that $0 \leq \delta \leq 1$. Balinski and Shahidi [5] concluded that the method $W$ with $\delta = \frac{1}{2}$ is simply the best method since it respects priorities, it is cyclic and equilibrated. We identify the two slightly new divisors with better performance in the next Section, to which we call mean-based divisors.

### 5.5 Mean-based Divisor Methods

The two mean-based divisor (MBD) functions are computed from available five divisors, both of which are based on arithmetic mean. The first divisor is calculated by the mean of all divisors whereas the second is calculated by the mean of Webster’s and Jefferson’s divisors. The first divisor falls between Hill’s and Webster’s divisors, immediate right to Hill’s divisor capturing the properties of both. Obviously the second one lies between Webster’s and Jefferson’s divisors. The mean-based divisor first (MBD1) is computed as

$$ d(a) = \frac{1}{5} \left[ a + \frac{2a(a+1)}{2a+1} + \sqrt{a(a+1)} + (a + \frac{1}{2}) + (a + 1) \right] $$

$$ = \frac{1}{5} \left[ 3a + \frac{3}{2} + \frac{2a(a+1)}{2a+1} + \sqrt{a(a+1)+3} \right] $$

$$ = \frac{16a^2 + 16a + 2(2a+1)\sqrt{a(a+1)+3}}{2(2a+1)} $$

$$ = \frac{\psi(a)}{10(2a+1)} $$

where $\psi(a) = (4a + 3)(4a + 1) + (4a + 2)\sqrt{a(a+1)}$.

The respective rank-index is defined by $r(p, a) = \frac{10p(2a+1)}{\psi(a)}$.
The mean-based divisor second (MBD2) is computed as

$$d(a) = \frac{1}{2} \left[ a + \frac{1}{2} + a + 1 \right] = a + \frac{3}{4}.$$  

The respective rank-index is defined by

$$r(p, a) = \frac{4p}{4a+3}.$$  

Clearly the second divisor is parametric with $$\delta = \frac{3}{4},$$ whereas the first divisor is not parametric. The actual location of our divisors among the existing five divisors can be sketched as follows: it is clear that divisor of $$W$$ is the middle point of the divisors of $$A$$ and $$J$$. Also the divisors of $$A, D, H$$ fall to the left of the divisor of $$W$$, being smallest number $$a$$, harmonic mean and geometric mean of $$a$$ and $$a+1$$ respectively, which is the fundamental rule of three means. The MBD1, being the arithmetic mean of divisors of $$A, D, H, W, J$$, clearly falls between the divisors of $$H$$ and $$W$$, nearer to $$H$$. The position of MBD2 is obvious. Hence, this inclusion can be stated as the following theorem [112].

**Theorem 5.3** The MBD1 lies between the divisors of $$H$$ and $$W$$, and MBD2 lies between the divisors of $$W$$ and $$J$$. That is, $$A < D < H < MBD1 < W < MBD2 < J.$$  

Thus, mean-based divisors are positioned in the neighborhood of divisors of $$H$$ and $$W$$. As the methods $$H$$ and $$W$$ are considered to be mathematically neutral with respect to emphasis on larger and smaller states, near to ideal fraction and consistent, the proposed new divisors yield the better results in apportioning the seats to states/parties with respect to population/votes. Moreover, they depend on all five divisors and hence preserve the qualities of all divisors at the best. With this discussion and in flavor of theorem 5.3, we establish the following theorem.

**Theorem 5.4** The mean-based divisors MBD1 and MBD2 generate the apportionments that are near to ideal, consistent, monotone and neutral.
In this sense, we claim that our divisors clearly point out the location of ideal apportionment, standing in the \textit{middle} of the divisors of other methods, and so they are better than others. However, we agree that time complexity of our methods is higher than other methods. The local measure of inequalities to be minimized under these two new divisors is considered with relative difference as given by Hill-Huntington, it is because the method yielding the smallest relative difference is taken as the best method. Together with our divisors, we argue that method of equal proportion is stable with the relative difference given by

$$T = \left| \frac{a_{ih}}{p_i} - \frac{a_{jh}}{p_j} \right| / \min \left( \frac{a_{ih}}{p_i}, \frac{a_{jh}}{p_j} \right) = \frac{a_{ih}p_j}{a_{jh}p_i} - 1$$ (5.6)

for $\frac{a_{ih}}{p_i} \geq \frac{a_{jh}}{p_j}$. The ideal position is $T = 0$ which is very rare in practice. Therefore, the measures of inequality between two states cannot be eliminated perfectly; and hence the fundamental objective is to minimize the measures of inequality as far as possible to reach as near as to the ideal position. To this point, we refer [111] for detail where a synthesized study has been carried out to minimize the inequality measures of JIT sequencing problem via apportionment approach.

\section{5.6 Efficient Frontiers}

In this Section, we deal both the problems simultaneously with similar objective functions establishing the necessary and sufficient condition for the optimality. The respective global and local deviation measures are compared as well. Furthermore, we prove the two propositions that we have recently proposed in [112] summing up with two corollaries.
5.6.1 The Global Deviation Approach

The global deviation measure means all the deviations of the actual and ideal productions (or apportionments) that are combined in some manner to form a single number which serves as a measure of fairness. Minimizing either of the two global indices \( \sum_{i=1}^{s} |a_{ih} - q_{ih}| \) or \( \sum_{i=1}^{s} (a_{ih} - q_{ih})^2 \) provides a sense of fair apportionment. The global index \( |x_{ik} - kr_i| \) of SDJIT sequencing problem in mixed-model systems is studied with oneness property in [27], focusing on the relation between optimal solutions of squared and absolute SDJIT sequencing problems (i.e., total PRVP). A sequence \( z \) is said to have oneness property if and only if \( -1 \leq z_{ik} \leq 1, \forall i, \forall k \). Considering absolute or squared deviation global indices, we can say that any sequence is feasible if the discrepancy of \( x_{ik} \) and \( kr_i \) lies between 0 and 1. Thus, main idea is to seek for the smaller bound nearer to 0. If \( x_{ik} = kr_i \), then it is done; however it is very rare in practice. In view of MBD1 defined in Subsection 5.4 for apportionment, we propose a stronger bound \( \beta^* \) for SDJIT sequencing problem based on arithmetic mean. Taking the intervals \([a, a + 1]\) and \([0, 1]\) together, the key idea is as follows [34]:

**Step 1.** Set \( a = 0 \) and \( a + 1 = 0 + 1 = 1 \).

**Step 2.** Make the partition of the interval \([0, 1]\) by computing harmonic mean \((0)\), geometric mean \((0)\) and arithmetic mean \((\frac{1}{2})\) of the two ends 0 and 1.

**Step 3.** Calculate the arithmetic mean of 0, 0, 0, \( \frac{1}{2} \), 1. That is, \( \beta^* = \frac{1}{5}(0 + 0 + 0 + \frac{1}{2} + 1) = \frac{3}{10} \).

To this end, we claim that \( \beta^* = 0.3 \) is the efficient bound near to ideal value yielding minimum deviation. The absolute and squared global deviation indices of both the total PRVP and DAP are equivalent and hence have same set of optimal solutions with the proposed bound \( \beta^* \). Moreover, we have
Theorem 5.5 The absolute SDJIT sequencing objective \( \sum_{i=1}^{n} \sum_{k=1}^{D} |x_{ik} - kr_i| \) and the absolute apportionment objective \( \sum_{i=1}^{s} |a_{ih} - q_{ih}| \) are equivalent and have same set of optimal solutions with the upper bound \( \beta^* \).

Proof: The notational correspondence between absolute SDJIT sequencing (i.e., total PRVP) and absolute apportionment objectives clearly shows the equivalency of the two problems. That is, product \( i \leftrightarrow \) state \( i \), \( n \leftrightarrow s \), \( d \leftrightarrow p \), \( d_i \leftrightarrow p_i \), \( k \leftrightarrow h \), \( x_{ik} \leftrightarrow a_{ih} \), \( kr_i \leftrightarrow q_{ih} \), \( \sum_{i=1}^{n} d_i \leftrightarrow \sum_{i=1}^{s} p_i \) theoretically imply that \( \sum_{i=1}^{n} \sum_{k=1}^{D} |x_{ik} - kr_i| \leftrightarrow \sum_{i=1}^{s} |a_{ih} - q_{ih}| \). Due to this equivalent nature, both the problems can be resolved by same rule and hence they have same set of optimal solutions with the efficient bound \( \frac{3}{10} \). The optimal solutions lie between the ideal value 0 and the bound 0.3. This completes the proof.

With the similar approach, we have the following theorem for squared deviation objectives.

Theorem 5.6 The squared SDJIT sequencing objective \( \sum_{i=1}^{n} \sum_{k=1}^{D} (x_{ik} - kr_i)^2 \) and the squared apportionment objective \( \sum_{i=1}^{s} (a_{ih} - q_{ih})^2 \) are equivalent and have same set of optimal solutions with the upper bound \( \beta^* \).

The complexity of total PRVP depends on \( n \) and \( D \), whereas the complexity of DAP depends on number of sates \( s \) and house size \( h \). Precisely, we have

Corollary 5.1 The complexities of total PRVP and DAP, that is, \( O(nD) \) and \( O(sh) \) respectively, are equivalent.
Clearly the other divisor MBD2, defined in Section 5.4 for apportionment, generates the larger bound $\frac{3}{4}$. Being parametric, we can say this bound generates cyclic JIT sequences, though it is not near to ideal.

### 5.6.2 The Local Deviation Approach

The local deviation measure means the pairwise minimization of inequalities between the production rates of any two products (or any two states) $i$ and $j$ that implies fairness and justice. According to the theory of equity measurement, the preference model should satisfy the principle of transfers which ensures that a transfer of small amount from an outcome to any relatively worse-off outcome results in a more preferred achievement vector. To this end, we restate Balinski and Shahidi’s objective to be minimized [5]:

$$\min_{x} \max_{i,j} \left| \frac{x_{ik}}{r_i} - \frac{x_{jk}}{r_j} \right|$$

(5.7)

For pairwise apportionment problem, we propose a similar objective to be minimized over the apportionment vector $a$ for the fairness measure between two states $i$ and $j$ as the local measure of inequality

$$\min_{a} \max_{i,j} \left| \frac{a_{ih}}{p_i} - \frac{a_{jh}}{p_j} \right|$$

(5.8)

We claim that the computational complexities of these two objectives are of same type, depending on the number of products and number of states respectively. But the complexities given in corollary 5.1 do not work here. In view of relative difference $T$ in equation (5.6) defined in Section 5.5 for apportionment, we define the relative difference $T^*$ for product to product rate variation problem and propose the equitably efficient (EE) solution as follows:

$$T^* = \left| \frac{x_{ik}}{r_i} - \frac{x_{jk}}{r_j} \right| / \min \left( \frac{x_{ik}}{r_i}, \frac{x_{jk}}{r_j} \right)$$

(5.9)

106
Theorem 5.7 If the state to state variation in the discrete apportionment problem is stable with relative difference $T$, then product to product rate variation in the JITSP is stable with relative difference $T^*$.

Proof: State to state variation problem (i.e., MEP) is mathematically neutral, near to ideal, monotone and consistent. So it suffices to show that state to state variation and product to product variation problems are mathematically equivalent. Products $i, j$ in JIT sequencing correspond with the states $i, j$ among $s$ states. The divisors $r_i$ and $r_j$ correspond with $p_i$ and $p_j$ in apportionment. The production rates of $i^{th}$ and $j^{th}$ products in period $k$ are equivalent to the shares of representatives of $i^{th}$ and $j^{th}$ states for house size $h$. Thus in view of MEP, we conclude that the product to product rate variation problem with the relative difference $T^*$ is near to ideal and mathematically neutral. This proves the theorem.

As an immediate consequence of theorem 5.7, we propose the following theorem and corollary:

Theorem 5.8 The state to state variation problem with $T$ has optimal solution if and only if the product to product rate variation problem with $T^*$ has optimal solution.

For an optimal production of a product, various parts should be produced in a balanced way. For example, all the parts of a car must be produced in balanced manner to produce a complete car. Otherwise, inventory of some parts and shortage of some other parts are generated. Similar case applies in the apportionment problem. Moreover, we have

Corollary 5.2 Balancing inter-state apportionments is equivalent to balancing sub-products of a product.
Thus we have made an attempt to correlate the single level mixed-model SDJIT sequencing problem with divisor-based DAP in proportional system. Our study presents the equivalency of sum deviation objectives of JIT and DAP as an efficient frontier for SDJIT sequencing problem \(i.e.,\) total PRVP in mixed-model production systems.

Recently, in a real-life benchmark study, Maier et al. [83] have studied DAP in terms of divisor-based biproportional apportionment as a two way proportionality electoral system, where the problem is to assign the integer number of seats to each party within each district such that the party requirements and district requirements are fulfilled addressing the issues of fairness and quality. Presenting some algorithms such as tie-and-transfer, alternating scaling and hybrid algorithms, the authors considered some benchmark instances mainly from the data of recent European elections. This approach also seems to be useful to handle SDJIT sequencing problem which is beyond of this dissertation. Nevertheless, this direction will certainly be an area to work in our future research.

In the apportionment context, we are interested to bring a brief note on the proportional representation (PR) system of the political parties in the Constitutional Assembly Election 2008, Nepal in the following Section. We obtained slightly new allocations of seats on effect of our MBD1 and MBD2. A brief case study in the Table 5.3 shows that the MBD1 is consistent, unbiased and mathematically neutral.

### 5.7 Proportional Representation: A Counter Example

There can be found several case studies on the different electoral systems around the world. We include here a brief record of recently performed constituional assembly election 2008 (CAE-08) in Nepal which used mixed
Thus we have made an attempt to correlate the single level mixed-model SDJIT sequencing problem with divisor-based DAP in proportional system. Our study presents the equivalency of sum deviation objectives of JIT and DAP as an efficient frontier for SDJIT sequencing problem (i.e., total PRVP) in mixed-model production systems.

Recently, in a real-life benchmark study, Maier et al. [83] have studied DAP in terms of divisor-based biproportional apportionment as a two way proportionality electoral system, where the problem is to assign the integer number of seats to each party within each district such that the party requirements and district requirements are fulfilled addressing the issues of fairness and quality. Presenting some algorithms such as tie-and-transfer, alternating scaling and hybrid algorithms, the authors considered some benchmark instances mainly from the data of recent European elections. This approach also seems to be useful to handle SDJIT sequencing problem which is beyond of this dissertation. Neverthless, this direction will certainly be an area to work in our future research.

In the apportionment context, we are interested to bring a brief note on the proportional representation (PR) system of the political parties in the Constitutional Assembly Election 2008, Nepal in the following Section. We obtained slightly new allocations of seats on effect of our MBD1 and MBD2. A brief case study in the Table 5.3 shows that the MBD1 is consistent, unbiased and mathematically neutral.

We computed the share of the parties using five divisor methods and mean-based divisor methods. To avoid the injustice, we set the notion that a party should receive at least 1 seat if its quota is greater than 0.5, since PR system should represent almost all groups of the people in the nation. This idea precisely reduced the seats of large parties and apportioned a seat for each additional 5 small parties (Table 5.3). Thus, 30 parties appear to be included in CAE-08 instead of 25. We claim that the MBD1 is the best divisor method.

The total number of votes casted for PR system is 1,07,39,078 and the standard divisor is \(\frac{1,07,39,078}{335} = 32,056\). Since the total number of votes dropped for PR system is considered, the whole population is represented by the seats of 30 parties. The seat in the 4th column of Table 5.3 means the allocated seats to the parties. The notations for the name of the parties in CAE-08 are abbreviated as follows:

CPN-M = Communist Party of Nepal (Maoist)
NC = Nepali Congress
UML = Communist Party of Nepal (Unified Marxist-Leninist)
MJAF = Madhesi Jana Adhikar Forum, Nepal
TMLP = Tarai-Madhesh Loktantrik Party
RPP = Rastriya Prajatantra Party
CPN-ML = Commmunist Party of Nepal (Marxist-Leninist)
Sad. P = Sadbhavana Party
JMN = Janamorcha Nepal
<table>
<thead>
<tr>
<th>Party</th>
<th>Votes Quota</th>
<th>Seats</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPN-M</td>
<td>31,448</td>
<td>98</td>
</tr>
<tr>
<td>CPN-U</td>
<td>22,696</td>
<td>70</td>
</tr>
<tr>
<td>UML</td>
<td>21,837</td>
<td>69</td>
</tr>
<tr>
<td>MJAF</td>
<td>6,783</td>
<td>21</td>
</tr>
<tr>
<td>TMLP</td>
<td>3,830</td>
<td>11</td>
</tr>
<tr>
<td>RPP</td>
<td>2,631</td>
<td>8</td>
</tr>
<tr>
<td>CPN-ML</td>
<td>2,435</td>
<td>8</td>
</tr>
<tr>
<td>Sad.P</td>
<td>1,677</td>
<td>5</td>
</tr>
<tr>
<td>JMN</td>
<td>1,641</td>
<td>5</td>
</tr>
<tr>
<td>CPN-U1</td>
<td>1,548</td>
<td>5</td>
</tr>
<tr>
<td>RPPN</td>
<td>1,019</td>
<td>3</td>
</tr>
<tr>
<td>RJM</td>
<td>646</td>
<td>3</td>
</tr>
<tr>
<td>RJP</td>
<td>477</td>
<td>3</td>
</tr>
<tr>
<td>NWPP</td>
<td>2,089</td>
<td>2</td>
</tr>
<tr>
<td>SLRM</td>
<td>1,958</td>
<td>2</td>
</tr>
<tr>
<td>NSP-A</td>
<td>1,671</td>
<td>2</td>
</tr>
<tr>
<td>RJP1</td>
<td>910</td>
<td>2</td>
</tr>
<tr>
<td>NJD</td>
<td>990</td>
<td>1</td>
</tr>
<tr>
<td>CPN-M1</td>
<td>600</td>
<td>1</td>
</tr>
<tr>
<td>DJP</td>
<td>348</td>
<td>1</td>
</tr>
<tr>
<td>NRP</td>
<td>757</td>
<td>1</td>
</tr>
<tr>
<td>SPJP</td>
<td>752</td>
<td>1</td>
</tr>
<tr>
<td>CBREP</td>
<td>575</td>
<td>1</td>
</tr>
<tr>
<td>NLSD</td>
<td>222</td>
<td>1</td>
</tr>
<tr>
<td>NPD</td>
<td>512</td>
<td>1</td>
</tr>
<tr>
<td>CPN-UM</td>
<td>717</td>
<td>1</td>
</tr>
<tr>
<td>LKJP</td>
<td>123</td>
<td>1</td>
</tr>
<tr>
<td>TNRD</td>
<td>657</td>
<td>1</td>
</tr>
<tr>
<td>RJD</td>
<td>305</td>
<td>0</td>
</tr>
<tr>
<td>CPN-UM</td>
<td>717</td>
<td>0</td>
</tr>
<tr>
<td>LKJP</td>
<td>123</td>
<td>0</td>
</tr>
</tbody>
</table>

Total: 1,073,9078

Table 5.3: Proportional Representation for 335 seats in CAE-08, Nepal
Table 5.3: Proportional Representation for 335 seats in CAE-08, Nepal

<table>
<thead>
<tr>
<th>Party</th>
<th>Votes</th>
<th>Quota</th>
<th>Seat</th>
<th>A</th>
<th>D</th>
<th>H</th>
<th>W</th>
<th>J</th>
<th>MBD1</th>
<th>MBD2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPN-M</td>
<td>31,44,204</td>
<td>98.08</td>
<td>100</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>NC</td>
<td>22,69,883</td>
<td>70.80</td>
<td>73</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>UML</td>
<td>21,83,370</td>
<td>68.10</td>
<td>70</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
</tr>
<tr>
<td>MJAF</td>
<td>6,78,327</td>
<td>21.16</td>
<td>22</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>RPP</td>
<td>2,63,431</td>
<td>8.21</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>CPN-ML</td>
<td>2,43,545</td>
<td>7.59</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Sad.P</td>
<td>1,67,517</td>
<td>5.22</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>RJM</td>
<td>1,64,381</td>
<td>5.12</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>CPN-U</td>
<td>1,54,968</td>
<td>4.83</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>CPN-U1</td>
<td>48,600</td>
<td>1.51</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>RPPN</td>
<td>1,10,519</td>
<td>3.44</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>RJP</td>
<td>1,02,147</td>
<td>3.13</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>NWPP</td>
<td>74,089</td>
<td>2.31</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SLRM</td>
<td>71,958</td>
<td>2.24</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NSP-A</td>
<td>55,671</td>
<td>1.73</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>RJP1</td>
<td>53,910</td>
<td>1.68</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NPD</td>
<td>48,990</td>
<td>1.52</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CPN-U1</td>
<td>48,600</td>
<td>1.51</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NRP</td>
<td>37,757</td>
<td>1.17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SPJP</td>
<td>35,752</td>
<td>1.12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CBREP</td>
<td>28,575</td>
<td>0.89</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>NLD</td>
<td>25,022</td>
<td>0.78</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>NPD</td>
<td>23,512</td>
<td>0.73</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CPN-M1</td>
<td>21,234</td>
<td>0.67</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TNRD</td>
<td>20,657</td>
<td>0.64</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RJD</td>
<td>19,305</td>
<td>0.60</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>CPN-UM</td>
<td>18,717</td>
<td>0.58</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>LKJP</td>
<td>18,123</td>
<td>0.56</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>1,07,39,078</td>
<td>335</td>
<td>335</td>
<td>335</td>
<td>335</td>
<td>335</td>
<td>335</td>
<td>335</td>
<td>335</td>
<td>335</td>
</tr>
</tbody>
</table>
Chapter 6

Chapter 6 is based on the papers B, D, K, M, O, Q, R.

In this chapter, we discuss on some problems that are associated with JITPS and on supply chain as a linkage of production with distribution problem. In Section 6.1, we present the five PRVP related problems. In Section 6.2, we give an overview of supply chain logistics relating with ORVP followed by the description of cross-docking operations for effective and efficient supply chain management. In Section 6.3, we give a bird's eye-view in performance modeling of queueing systems as one of our recent works.

6.1 PRVP Related Problems

A significant number of real-life problems related with PRVP do exist in the literature. We have given the mathematical linkage of PRVP with discrete apportionment in Chapter 5. In the following Subsections, we discuss the five PRVP related problems, namely response time variability, Liu-Layland periodic scheduling, pinwheel scheduling, fair queueing and JIT delivery.
Chapter 6

PRVP Related Problems, Supply Chain and Queueing

Chapter 6 is based on the papers B, D, K, M, O, Q, R.

In this chapter, we discuss on some problems that are associated with JITPS and on supply chain as a linkage of production with distribution problem. In Section 6.1, we present the five PRVP related problems. In Section 6.2, we give an overview of supply chain logistics relating with ORVP followed by the description of cross-docking operations for effective and efficient supply chain management. In Section 6.3, we give a bird’s eye-view in performance modeling of queueing systems as one of our recent works.

6.1 PRVP Related Problems

A significant number of real-life problems related with PRVP do exist in the literature. We have given the mathematical linkage of PRVP with discrete apportionment in Chapter 5. In the following Subsections, we discuss the five PRVP related problems, namely response time variability, Liu-Layland periodic scheduling, pinwheel scheduling, fair queueing and JIT delivery.
6.1.1 Response Time Variability Problem

The response time variability problem (RTVP) is a combinatorial optimization problem, recently developed in [28] to model a broad range of real-life situations. The RTVP occurs when customers, jobs, events or products are required to be sequenced to minimize the variability of the time that they wait for their next turn in obtaining the resources they need to advance. More concisely, RTVP is designed to minimize variability in the distance between any two consecutive copies of the same models.

This problem minimizes the variability of the time that a material or a product waits for its next turn to be sequenced to form a fair production sequence. Clearly, this problem is based on the distance but not on the position of those products with only one copy to be sequenced [29]. However, the bottleneck PRVP is based on either the variation between ideal and actual production [103] or the variation between ideal and actual time of production [56].

Now we formulate the RTVP as follows: Let \( n \) be the number of products or jobs, \( d_i \) be the number of copies of the model \( i \) to be scheduled (\( i = 1, 2, \ldots, n \)) and \( D \) be the total number of copies such that \( D = \sum_{i=1}^{n} d_i \). Let \( s \) be a solution of an instance of RTVP consisting of a circular sequence of copies \( s = s_1 s_2 \ldots s_D \), where \( s_j \) is the copy sequenced in position \( j \) of sequence \( s \). For each model \( i (d_i \geq 2) \), let \( t_{ik} \) be the distance between the positions in which copies \( k + 1 \) and \( k \) of model \( i \) are found, where the distance between two consecutive positions is considered to be 1. Since \( s \) is circular, position 1 comes immediately after position \( D \); so \( t_{id_i} \) is the distance between the first copy of model \( i \) in a cycle and the last copy of the same model in the preceding cycle.

Let \( \bar{t}_i \) be the desired average distance between two consecutive copies of model \( i \) such that \( \bar{t}_i = \frac{D}{d_i} \). Note that for each model \( i \) in which \( d_i = 1, t_{i1} \) is equal to \( \bar{t}_i \). The goal is to minimize the metric, called response time variability
Response Time Variability Problem (RTVP) is a combinatorial optimization problem, recently developed in [28] to model a broad range of real-life situations. The RTVP occurs when customers, jobs, events or products are required to be sequenced to minimize the variability of the time that they wait for their next turn in obtaining the resources they need to advance. More concisely, RTVP is designed to minimize variability in the distance between any two consecutive copies of the same models. This problem minimizes the variability of the time that a material or a product waits for its next turn to be sequenced to form a fair production sequence. Clearly, this problem is based on the distance but not on the position of those products with only one copy to be sequenced [29]. However, the bottleneck PRVP is based on either the variation between ideal and actual production [103] or the variation between ideal and actual time of production [56].

Now we formulate the RTVP as follows: Let $n$ be the number of products or jobs, $d_i$ be the number of copies of the model $i = 1, 2, ..., n$, and $D$ be the total number of copies such that $D = \sum_{i=1}^{n} d_i$. Let $s$ be a solution of an instance of RTVP consisting of a circular sequence of copies ($s = s_1 s_2 ... s_D$), where $s_j$ is the copy sequenced in position $j$ of sequence $s$. For each model $i$ ($d_i \geq 2$), let $t_{ik}$ be the distance between the positions in which copies $k + 1$ and $k$ of model $i$ are found, where the distance between two consecutive positions is considered to be 1. Since $s$ is circular, position 1 comes immediately after position $D$; so $t_{id}$ is the distance between the first copy of model $i$ in a cycle and the last copy of the same model in the preceding cycle.

Let $\bar{t}_i$ be the desired average distance between two consecutive copies of model $i$ such that $\bar{t}_i = D/d_i$. Note that for each model $i$ in which $d_i = 1$, $t_{i1}$ is equal to $\bar{t}_i$. The goal is to minimize the metric, called response time variability (RTV), defined by the sum of the squared deviations with respect to the $\bar{t}_i$ distances. This objective is defined by the following non-linear expression for model $i$ [28]:

$$\text{minimize} \left[ RTV = \sum_{i=1}^{n} \sum_{k=1}^{d_i} (t_{ik} - \bar{t}_i)^2 \right] \quad (6.1)$$

The RTV metric (6.1) is a weighted variance with weights equal to $d_i$; which is expressed as

$$\text{RTV} = \sum_{i=1}^{n} d_i \ast \text{Var}_i, \text{ where } \text{Var}_i = 1/d_i \sum_{i=1}^{d_i} (t_{ik} - \bar{t}_i)^2.$$

It is established that a solution procedure that minimizes the bottleneck product rate variation problem with $n = 2$ also minimizes the response time variability problem for two product case [28, 60].

It is shown that the complexity of the RTVP is NP-hard [28] and so it is difficult to be solved optimally. The reduction of NP-hardness of RTVP is from the periodic maintenance scheduling problem shown NP-complete by Bar-Noy et al. [10]. The useful mathematical model to solve the response time variability problem with large instance size is presented in [29] and the near-optimal solution with a genetic algorithm is described in [125]. Some heuristics and meta-heuristics are used to solve RTVP as recent approaches for which timely information flow is a determining factor.

Five heuristics are proposed to solve the RTVP in [28], namely Bottleneck algorithm to solve the minmax PRVP, Random generation, Two classical parametric methods for solving the apportionment problem called Webster method and Jefferson method [5], a new heuristic called Insertion method. Moreover, a local search procedure is applied to the solutions obtained with the five heuristics. For detail explanation, we refer [73].
6.1.2 Liu-Layland Periodic Scheduling

The Liu-Layland periodic scheduling problem is one of the fundamental problems in hard real-time computing systems [73], which was initially introduced by Liu and Layland in [81]. A system is said to be real-time if its operation depends on logical correctness of jobs of the system and on the time as well. The hard real-time problems means that the completion of job after its dead line is extremely meaningless. The whole system may be failure due to this delay and a huge loss can happen, such as in aircraft scheduling.

A relation between bottleneck PRVP and Liu-Layland problem has been established in [70, 73] presenting a correspondence between the two problems. Furthermore, apportionment methods exist in the literature [59] for the solution to the Liu-Layland periodic scheduling problem. Not all the divisor methods work for this, but slight modification of Adam’s and Jefferson’s divisor methods solve the Liu-Layland problem resulting into the quota-Adams and the quota-Jefferson divisor methods introduced in [6, 106]. These particular quota methods are conceptually the simplest and computationally the most efficient ones among all quota divisor methods. A necessary condition for any divisor method to solve the Liu-Layland problem is to stay above the lower quota. Similarly, staying below upper quota is a necessary condition for any divisor method to solve the Liu-Layland problem [73].

Now we discuss the mathematics of periodic sequencing problem as follows [81]: Consider $i$ (i = 1, 2, ..., n) periodic jobs that are independent and preemptive. Let the request period and run-time of the jobs be $R_i$ and $\bar{R}_i$ respectively such that $\bar{R}_i \leq R_i$; $i = 1, 2, \ldots, n$. The preemption occurs at integral points only. Hard-real-time means the requested job $i$ at moment $kR_i$ must be executed by the moment $(k + 1)R_i$. The deadlines $R_i, 2R_i, \ldots$ are considered hard for job $i$.

The Liu-Layland periodic scheduling problem is to find an infinite sequence
6.1.2 Liu-Layland Periodic Scheduling

The Liu-Layland periodic scheduling problem is one of the fundamental problems in hard real-time computing systems [73], which was initially introduced by Liu and Layland in [81]. A system is said to be real-time if its operation depends on logical correctness of jobs of the system and on the time as well. The hard real-time problems means that the completion of job after its deadline is extremely meaningless. The whole system may be failure due to this delay and a huge loss can happen, such as in aircraft scheduling.

A relation between bottleneck PRVP and Liu-Layland problem has been established in [70, 73] presenting a correspondence between the two problems. Furthermore, apportionment methods exist in the literature [59] for the solution to the Liu-Layland periodic scheduling problem. Not all the divisor methods work for this, but slight modification of Adam’s and Jefferson’s divisor methods solve the Liu-Layland problem resulting into the quota-Adams and the quota-Jefferson divisor methods introduced in [6, 106]. These particular quota methods are conceptually the simplest and computationally the most efficient ones among all quota divisor methods. A necessary condition for any divisor method to solve the Liu-Layland problem is to stay above the lower quota. Similarly, staying below upper quota is a necessary condition for any divisor method to solve the Liu-Layland problem [73].

Now we discuss the mathematics of periodic sequencing problem as follows [81]: Consider \( s = s_1s_2 \ldots \) on a finite sequence \( \{1, 2, \ldots, n\} \) of \( n \) in a single machine such that \( s_j \in \{1, 2, \ldots, n\}, j \in N \). The job \( i \) occurs exactly \( \bar{R}_i \) times on any subsequence of \( s \), consisting of \( R_i \) consecutive elements of \( s \) with \( \bar{R}_i \leq R_i \), \( i \in \{1, 2, \ldots, n\} \) [70].

The deadline driven algorithm of Liu and Layland [81] assigns priorities to tasks according to the deadlines of their current requests. Therefore, a task with the highest priority at a unit time slot \( k \) will be the one with the deadline of its current request being the nearest to \( k \), and a task will be assigned the lowest priority if the deadline of its current request is the furthest from \( k \). In any unit time slot \( k \), a task with the highest priority and still incomplete current request will be executed [73].

The bottleneck product rate variation problem with bottleneck less than 1 is periodic where the demand rates are \( r_i = \frac{\bar{R}_i}{R_i}, i = 1, 2, \ldots, n \) and \( \gcd(\bar{R}_1, \bar{R}_2, \ldots, \bar{R}_n) > 1 \). A necessary and sufficient condition for the jobs 1, 2, \ldots, \( n \), in the Liu-Layland scheduling problem to be periodic is \( \sum_{i=1}^{n} \frac{\bar{R}_i}{R_i} \leq 1 \) [81]. This shows that any sequencing procedure to the bottleneck PRVP is also a sequencing procedure to the Liu-Layland scheduling periodic problem [70]. It has been shown that any period \( [(k - 1)R_i + 1, kR_i] \) of a sequence of the bottleneck product rate variation problem consists of at least \( \bar{R}_i \) copies of model \( i \).

Thus, we can say that the Liu-Layland periodic scheduling problem has connection with PRVP and discrete apportionment problem.

6.1.3 Pinwheel Scheduling Problem

With the motive to save satellite-data, the pinwheel scheduling problem was introduced for the real-time satellite communication with a ground station without data loss [54, 73]. The ground station receives data from a number
of satellites in this problem. In each time slot, the station can only receive a single data packet from a single satellite. Each satellite has possibly different window of time of length $b$, measured in the number of time slots to broadcast repeatedly the same data packet to the ground station. The station then has to allocate time slots to satellites so as to ensure that no data packet is lost, that is a satellite with time window $b$ must be allocated at least one time slot out of any consecutive $b$ time slots.

A generalized pinwheel scheduling problem was introduced in [11, 12]. This problem requires a sliding time window of size $b$ that always includes at least $a \geq 1$ time slots allocated to a satellite, which can speed up error recovery in case of possible broadcast problems. Next we define pinwheel scheduling problem and its generalized case below.

The pinwheel scheduling problem is to find a pinwheel schedule as an infinite sequence $s = s_1 s_2 \ldots$ on the set $\{1, 2, \ldots, n\}$ for given $b_i, i = 1, 2, \ldots, n$ such that i) $s_j \in \{1, 2, \ldots, n\}, \forall j \in N$ and ii) Each $i \in \{1, 2, \ldots, n\}$ occurs at least once in any subsequence of $s$ consisting of $b_i$ consecutive elements of $s$. We refer [73] for detail explanation.

The generalised pinwheel scheduling problem for given pairs of positive integers $(a_1, b_1), (a_2, b_2) \ldots, (a_n, b_n)$ is to find an infinite sequence $s = s_1 s_2 \ldots$ on a finite sequence $\{1, 2, \ldots, n\}$ such that $s_j \in \{1, 2, \ldots, n\}, j \in N$ and any subsequence of $s$ consisting of $b_i$ consecutive elements of $s$ contains $i$ at least $a_i$ times, $i = 1, 2, \ldots, n$. The generalized pinwheel scheduling problem is NP-hard [73].

An optimal sequence $s$ of an instance with the demand rate $r_i = \frac{a_i + 1}{b_i}$ to the bottleneck PRVP consists of at least $a_i$ copies of model $i, i = 1, 2, \ldots, n$, in any subsequence of $b_i$ consecutive elements of $s$. So, $\sum_{i=1}^{n} \frac{a_i}{b_i} + \frac{1}{b_i}$ is the necessary and sufficient condition that an instance with the demand rate $r_i = \frac{a_i + 1}{b_i}$ to the bottleneck PRVP has a generalized pinwheel schedule. The
generalized pinwheel schedule always exists for $n = 2$. For more detailings and applications of generalized pinwheel scheduling problem, we refer [73].

6.1.4 Fair Queueing

The performance of packet switched data networks is greatly influenced by the queue service discipline in routers and switches. The fair queueing discipline has several advantages over the traditional first-come-first-served discipline. The fair queueing algorithms set the rules of bandwidth allocation at the gateways of the network so that a good user behavior is encouraged and enforced throughout the network [73]. These algorithms are crucial in congestion control where bandwidth and buffer space are allocated fairly. More precisely, the queueing algorithms rely on three quantities—bandwidth (which packets get transmitted), promptness (when do those packets get transmitted) and buffer space (which packets are discarded by the gateway).

The fair queueing algorithms, proposed in the literature [73, 93] and implemented in practice, to determine the rules of the gateways bandwidth allocation are essentially two basic parametric methods of apportionment—the Adams’s and the Jefferson’s methods. The authoritarian solution has perhaps unknowingly been found in the apportionment solution to the fair representation problem of meeting the ideal of one man, one vote [6].

The two queueing algorithms have been formulated in the literature as follows [73, 93]:

1. Fair queueing based on starting times stipulates that whenever a packet finishes transmission on outgoing link the next one transmitted on that link is the one with the earliest start time.

2. Fair queueing based on finishing times stipulates that whenever a packet finishes transmission on outgoing link the next one transmitted on that
link is the one with the earliest finish time.

To describe mathematically, let the packet size of queue \( i \) be \( \rho_i \) and the number of packets from the queue \( i \) that got transmitted on the outgoing link be \( a_i \). Whenever a packet from some queue finishes transmission, then the first algorithm selects a next packet from the queue (flow) \( i^* \) such that

\[
\rho_j a_j \geq \rho_{i^*} a_{j^*}, \quad \forall j
\]

whereas the second algorithm selects a next packet from the queue \( i^* \) such that

\[
\rho_j (a_j + 1) \geq \rho_{i^*} (a_{j^*} + 1), \quad \forall j
\]

These algorithms show that the fair queueing based on starting times is clearly the Adams’s method of apportionment and the fair queueing based on finishing times is the Jefferson’s method (see Chapter 5 and Table 5.2 for detail), where the population of state \( i \) is \( \frac{1}{\rho_i} \). Clearly, the fair queueing based on starting times maximizes the minimum share of the bandwidth, that is, it optimizes

\[
\max \min_i \frac{a_i}{p_i} = \max \min_i \rho_i a_i
\]

whereas the fair queueing based on finishing times minimizes the maximum share of the bandwidth, that is it optimizes

\[
\min \max_i \frac{a_i}{p_i} = \min \max_i \rho_i a_i
\]

Furthermore, the fair queueing based on the midpoints resulting in the Webster’s method of apportionment, provides a number of advantages over the Adams’s and the Jefferson’s methods. The fair queueing based on the midpoint selects a next packet from the queue \( i^* \) such that

\[
\rho_j \left( a_j + \frac{1}{2} \right) \geq \rho_{i^*} \left( a_{i^*} + \frac{1}{2} \right), \quad \forall j.
\]
The Websters method minimizes the weighted squared difference between each queue share and the ideal share defined as follows [73]:

$$\text{minimize } \sum_i p_i \left( \frac{a_i}{p_i} - \frac{a}{P} \right)^2 \quad (6.6)$$

where the queue $i$ share of the bandwidth is $\frac{a_i}{p_i} = \rho_i a_i$ and the ideal share for all the queues at the gateway is $\sum \frac{a_i}{p_i} = \frac{a}{P} = \frac{\rho}{P}$.

These three methods, the Adams’s and the Jefferson’s and the Webster’s, for the fair queueing algorithm, are the parametric methods of apportionment and hence they are all uniform. The max-min fairness criterion, relative fairness bound and absolute fairness bound are subsequently described in [73].

### 6.1.5 Just-in-Time Delivery

The just-in-time (JIT) delivery is an inventory control system that has been the preferred method for meeting the two goals of the firms - reducing inventory and avoiding delays. This means that a supplier must be able to provide JIT delivery - reliably getting products there just before the customer needs them. That is, the materials are provided to the production plant just as they are required for use and end products are delivered to the final customers at the desired time. Most of the manufacturing companies which previously carried huge finished goods inventories to meet customers demand, have implemented the JIT delivery system so that the lot sizes are reduced and the frequency of delivery is scheduled as per customers demand.

JIT delivery is part of the larger concept of JIT purchasing that includes a small and reliable supplier base close to the buyers plant and frequent deliveries. A smooth flow of materials is required between suppliers and buyers as one of the key elements needed to ensure a continuous process.
from the receipt of raw material/components through to the shipment of the finished goods [100]. Therefore, JIT delivery is concerned with the issues of transportation and logistics. The success of JITPS largely relies on the effective JIT delivery systems.

With the increase of online-shopping (i.e., home deliveries), small scale orders and deliveries present new logistical challenges for all partners including more non-stop logistic movement, such as: cross-docking, consolidation of products from multiple manufacturers by third-party logistics providers in a single delivery, increased emphasis on point-of-sale driven, pull inventory replenishment systems, increased demand for customized deliveries of multi-tier pallets with electronic pallet content identification, advanced electronic data interchange capabilities.

In addition of delivering goods to customers in JIT delivery, the manufacturer should acquire accurate data of the customers’ demand and maintain an optimum schedule to coordinate the distribution system and hence to balance whole supply chain systems. In order to coordinate the production with customers’ demand in fixed time intervals, and to synchronize the ordering of raw materials with production schedules, both raw materials and finished items inventories should be maintained at an economic level to reduce the total cost of the system.

The JIT delivery (i.e., JIT distribution) is the root idea for the evolution of broader concept- the supply chain management (SCM). That is, the JITPS movement of 1980s evolved into SCM in 1990s [50], which is an integral part of supply chain logistics today. Both JITPS and SCM encourage to reduce the number of suppliers, to build relationships for the long term, to convey critical information on both long and short-term production plans in order to maximize companys effectiveness. However, a key difference between JITPS and SCM lies in their perspectives. The SCM perspective is broader than the perspective of JITPS. The former adopts a systematic view of the supply
from the receipt of raw material/components through to the shipment of the finished goods [100]. Therefore, JIT delivery is concerned with the issues of transportation and logistics. The success of JITPS largely relies on the effective JIT delivery systems.

With the increase of online-shopping (i.e., home deliveries), small scale orders and deliveries present new logistical challenges for all partners including more non-stop logistic movement, such as: cross-docking, consolidation of products from multiple manufacturers by third-party logistics providers in a single delivery, increased emphasis on point-of-sale driven, pull inventory replenishment systems, increased demand for customized deliveries of multi-tier pallets with electronic pallet content identification, advanced electronic data interchange capabilities.

In addition of delivering goods to customers in JIT delivery, the manufacturer should acquire accurate data of the customers’ demand and maintain an optimum schedule to coordinate the distribution system and hence to balance whole supply chain systems. In order to coordinate the production with customers’ demand in fixed time intervals, and to synchronize the ordering of raw materials with production schedules, both raw materials and finished items inventories should be maintained at an economic level to reduce the total cost of the system.

The JIT delivery (i.e., JIT distribution) is the root idea for the evolution of broader concept - the supply chain management (SCM). That is, the JITPS movement of 1980s evolved into SCM in 1990s [50], which is an integral part of supply chain logistics today. Both JITPS and SCM encourage to reduce the number of suppliers, to build relationships for the long term, to convey critical information on both long and short-term production plans in order to maximize company’s effectiveness. However, a key difference between JITPS and SCM lies in their perspectives. The SCM perspective is broader than the perspective of JITPS. The former adopts a systematic view of the supply chain.

The integrated philosophy of SCM is complemented by the activity-based philosophy of JITPS. The JITPS may benefit certain functional areas of a company through optimization of inventory, while other non-targeted functions of a company are not touched upon. The SCM in going beyond optimizing performance at a company and integration operations and elimination waste at suppliers, suppliers’ supplier, customer, customer’s customer [126]. Generally, supply chain is viewed as a set of relationships wherein a number of various business entities work together to perform the procurement of raw material, its transformation into intermediate and end-products for the customers. It is traditionally characterized by forward flow of materials and a backward flow of information. We discuss further details of supply chain logistics with cross-docking operations in the following Section.

6.2 Supply Chain

Production and distribution processes are like the two faces of a coin; unless there is effective and efficient distribution system, the production face has no meaning only generating inventories in each level of production. It is, therefore, integration of overall production and distribution processes within and among companies is being the most recent fundamental trend in the domain of production management. A synchronized view of these processes consists of all traditional areas of supplier-buyer relationships coping with the production of goods and their distribution as well (See Figure 3.2 too), in particular production and logistics [113, 116]. The Figure 6.1 pictorizes the logistical networks in supply chain system from upstream management to downstream management. The overall problem considering all the levels at a time is really a difficult one. The final assembly line problem as single-level product sequencing is best researched in the literature. However, the supply
chain logistics along the multi-level sequencing problem has enough space to be researched further. From its structural nature, the inbound logistics problem is more difficult and less researched whereas the outbound logistics problem have been dealt by many scientific researchers.

A sequenced delivery of the raw materials, parts and products throughout overall supply chain systems of manufacturing companies is the ultimate realization of JIT principles - zero inventories, zero defects and zero waste. In an ideal JIT production and distribution environment, parts should be delivered to the workstations at the exact time they are needed, in the exact quantity required, and in exact price and quality demanded. This type of just-in-time movement keeps the overall supply chain systems in a balanced condition. This situation requires the update co-ordination and real time information flow along all the nodes of supply chain production system.

![Figure 6.1: Logistics networks in supply chain](image)
6.2.1 Supply Chain Logistics

A typical supply chain consists of raw material suppliers, manufactures, distributors, retailers and end customers. Raw materials are shipped to the production facility where they get converted to end products and then those end products are shipped to the end users (customers). In order to minimize cost and waste throughout the system, effective supply chain management and supply chain integration are required beginning from the raw materials and ending with the final customers.

The supply chain logistics is the task of integrating organizational units along a supply chain and coordinating materials, information and financial flows to fulfill the final customer demands with the aim of improving competitiveness of the manufacturing company as a whole. The production logistics indicates the entire process of materials, data and products moving into, through, and out of a manufacturing firm. Inbound logistics covers the movement of materials received from initial suppliers, material management describes the movement of materials and components within a firm, and outbound logistics refers the physical distribution of finished goods from the end of assembly line to the customers [114]. To realize the best quality production and timely distribution for the customer in a rapidly changing technical environment, it is essential to create a cross-docking environment throughout the whole supply chain system that is capable to address the diversified demands [113].

Cross-docking is the movement of products directly from the receiving dock to the shipping dock with minimum dwell time in between. By arranging for immediate cross-docking of incoming products, retailers are able to reduce to a minimum in-transit time for their incoming products. Moreover, cross-docking is a logistics technique used in the retail and trucking industries with operations seeking to move materials from inbound locations to outbound locations as quickly as possible. However, cross-docking operations require good information systems and close synchronization of all
inbound and outbound shipments. The mutual coordination among all the independent firms (viz., raw-material suppliers, manufacturers, distributors and retailers) is the crux to attain the flexibility required to enable them in the progressive improvement of logistics processes in response of rapidly changing market conditions.

6.2.2 Cross-Docking Operations

As a dynamic JIT distribution centre, cross-docking has been widely applied in both manufacturing systems and logistics, since cross-docking operation (CDO) favors the timely distribution of freight and a better synchronization with the demand to provide the improved customer service. For instance, perishable products reach the marketplace faster preserving quality and freshness. CDO is considered as the best method to reduce inventory and to improve responsiveness of various customer demands. It is a process where products are received in a distribution center occasionally merged with other products going to the same destination, then shipped at the earliest opportunity.

A cross-docking supply chain logistics (CDSCL) system is a material handling and distribution concept in which the products move directly from the receiving dock to the shipping dock without being stored in a warehouse or distribution center [113]. The sortation and consolidation of the products at the transfer terminal is shown in Figure 6.2 and the operational scenario at cross-docking center is presented in Figure 6.3. The logistics cost could be reduced by integrating the inbound and outbound vehicles in the distribution system. A framework for understanding and designing cross-docking systems is provided in [115, 124] discussing the techniques that can improve the overall efficiency of the logistics and distribution operations. Only few research papers deal with the short-term scheduling problems arising from the daily operations of cross-docking terminals. Material handling inside the
cross-dock terminal for a given truck schedule is considered (see [113] for sufficient references). The advantages of CDO include minimal inventory, low handling costs, low space requirement, centralized processing and low transportation costs. Once a set of inbound and outbound trucks is docked, jobs consisting of products to be handled have to be assigned to resources, that is, workers and means of conveyance like fork-lifts are handled in such a way that efficient unloading, sorting and loading operations render possible. It is modeled as a machine scheduling problem and proposed a meta-heuristic suited for its solution [113]. A specific truck scheduling problem is covered at a parcel hub and solved by a simulation-based optimization approach. A special kind of cross-dock terminal with a conveyor belt system is treated in [129] where the transportation of goods within the dock is modeled as a
detailed scheduling problem providing a priority rule based start heuristic. Boysen et al. [21] treated a stylized one inbound door serves one outbound door setting in order to generate fundamental insights to the underlying real-world problem structure. Exact handling times for inbound trailers depend on the exact packing of goods and the sequence in which they can be obtained, whereas those for outbound trailers have to account for load stability and the sequence in which customers are served. Furthermore, the determination of transportation times between doors results to a complex optimization problem in itself. Thus, handling times used in a detailed truck

Figure 6.3: Operational scenario at cross-docking center (CDC)
scheduling model are merely estimated average times and often bound to heavy inaccuracies. Under such pre-requisites, detailed models may lead to more misleading or even infeasible plans when compared to aggregate models. So individual handling times for products are merged to service slots to which inbound and outbound trucks are assigned. A slot comprises the time required for completely unloading an inbound truck and completely loading an outbound truck respectively. Handling times in between dock doors are considered by a delay which covers the time span until incoming products are available at an outbound door. By a simultaneous scheduling of inbound and outbound trucks, incoming flows of products are harmonized with outbound flows, so that a JIT supply of products, and thus, a reduced turnover time is enabled.

6.2.3 The CDSCL Model Description

The cross-docking supply chain logistics (CDSCL) problem is considered as truck sequencing problem (TRSP) over here. The notational convention is described as follows: Let $I$ and $O$ be the sets of inbound and outbound trucks at the single receiving door and the single shipping door respectively of the cross-docking terminal. Each inbound truck is loaded with units of different products $p \in P$. Suppose $a_{\alpha p}$ be the number of units of product type $p$ arriving in an inbound truck $\alpha$, and $b_{\beta p}$ be the number of product type $p$ to be loaded onto outbound truck $\beta$.

All product units are completely unloaded within a service slot $t$ to which the respective inbound truck is assigned, so that all handling operations (e.g., docking, unloading and undocking) required to process the truck are executed within this time span, e.g., an hour or two. Moreover, all inbound trucks are assumed to be available for processing at the beginning of the planning horizon, so that a static problem with identical arrival dates of inbound trucks is considered. The assumption of equally long service times can be seen as
a reasonable approximation of reality, whenever vehicle capacities and the
count of products per vehicle do not strongly differ. As trailers are typically
of a standardized size and cross-docking aims at moving only full truck loads,
this premise is fulfilled whenever all processed products are of comparable size
(e.g. mail distribution centers) or all truck loads resemble a representative
average truck load (e.g., rotational deliveries of special promotional offers to
all stores of a retail chain).

Once unloaded, the delivered products have to undergo several subsequent
operations before they are available for being loaded onto the outbound
trucks at the shipping door. These operations include recording of any prod-
cuct unit in the information system, examining the product correctness and
quality, collecting, sorting, rearranging and packing to recombine products
from different inbound trucks to form the load of a certain outbound truck.
Finally, the products have to be transported to the shipping door, where
they wait in an intermediate buffer of sufficient size until they are needed.
This variety of tasks from recording to transporting is assumed to last a fixed
movement time $m$ irrespective of the truck load actually processed. Then,
al products arriving in a slot $t$ are available for loading at the shipping dock
not before slot $t + m$ if the movement process can be started for any un-
loaded unit immediately, *e.g.*, when applying a conveyor belt system. If the
movement starts not before the complete inbound truck has been unloaded
completely (*e.g.*, a worker stacks all units behind the receiving door before
moving them), the units are first available at slot $t + m + 1$. However, the
displacement $m$ or $m+1$, respectively, can be ignored (set to zero) when mod-
eling and solving the problem, because after having determined a solution,
an appropriate re-indexing of slots outbound trucks are assigned to allow
the exact determination of the outbound schedule. Similarly to constant
unloading times, it is assumed that the movement time $m$ is independent
of the inbound truck and the loaded products, because handling full truck
loads, which may always consist of almost the same number of product units,
should take very similar times. This assumption is realistic especially within an aggregated medium-term scheduling approach.

The set $O$ of outbound trucks is to be loaded at the shipping door for each $eta \in O$ with a predetermined number of units $b_{\beta p}$ of the different products $p \in P$. Also, it is assumed that all handling operations per truck are completed within a single slot. An outbound truck can be assigned to a slot $t$ not before enough stock has accumulated in the intermediate buffer to serve all demanded product units of the truck. As only temporary stock is allowed (or desired) within a cross dock, it is assumed that temporary stock is empty before the first inbound truck arrives and is emptied out again after the last outbound truck was served. This implies that the following equality holds within the model of the problem:

\[
\sum_{\alpha \in I} a_{\alpha p} = \sum_{\beta \in O} b_{\beta p}, \quad \forall p \in P
\]  

(6.7)

The simplifying assumptions applied to the base model of CDSCL problem are as follows [21]:

1. Inbound trucks are processed at a single receiving door of the terminal that serves a single shipping door for outbound trucks. Both doors are distinct giving segregated mode of service.

2. Processing trucks (i.e., all loading and unloading operations) takes the same amount of time.

3. No predefined restrictions on truck assignments to slots (e.g., release or due dates) exist.

4. The input data is known in advance with certainty (static deterministic problem).

5. The movement time of products across the dock is a given constant and so it can be ignored.

131
The sum of units delivered by inbound trucks equals the sum of units consumed by outbound trucks for any product $p$ (only intermediate stock).

Intermediate buffer for intermediate stock is not limited in size.

### 6.2.4 The CDSCL Model Formulation

The TRSP has been modeled in various forms as the supply chain logistics problem. Here, we set the following notations to formulate the CDSCL problem [21, 113]:

- $I$ = set of inbound trucks (indexed by $\alpha$)
- $O$ = set of outbound trucks (indexed by $\beta$)
- $P$ = set of products (indexed by $p$)
- $T$ = total number of time slots (indexed by $t$)

$a_{\alpha p}$ = quantity of product $p$ arriving in truck $\alpha$

$b_{\beta p}$ = quantity of product $p$ to be loaded in truck $\beta$

$x_{\alpha t} = 1$, if inbound truck $\alpha$ is assigned to slot $t$, and 0, if otherwise

$y_{\beta t} = 1$, if outbound truck $\beta$ is assigned to slot $t$, and 0, if otherwise

We define operation time of $\alpha$ trucks by $tx_{\alpha t}$ and operation time of $\beta$ trucks by $ty_{\beta t}$. As a direct result of the above simplifying assumptions (see [21] also), the inbound and outbound schedule can be readily derived by the sequence of inbound and outbound trucks, so that the problem reduces to TRSP. The objective is to sequence the trucks in such a way that the operation time is minimized which comprises the time span starting from the first slot to which an inbound truck is assigned and lasts until the final slot in which an outbound truck is processed. With the above discussions and notations, we formulate the TRSP to minimize discrepancies of the operation times of
outbound and inbound trucks as follows:

\begin{align*}
\text{Minimize} \quad & M = \max |tx_{\alpha t} - ty_{\beta t}|, \quad \forall \alpha \in I \text{ and } \beta \in O \quad (6.8) \\
\text{subject to} \quad & \sum_{t=1}^{T} x_{\alpha t} = 1, \quad \forall \alpha \in I \quad (6.9) \\
& \sum_{\alpha \in I} x_{\alpha t} \leq 1, \quad \forall t = 1, 2, \ldots, T \quad (6.10) \\
& \sum_{\beta \in O} y_{\beta t} = 1, \quad \forall \beta \in O \quad (6.11) \\
& \sum_{\beta \in O} y_{\beta t} \leq 1, \quad \forall t = 1, 2, \ldots, T \quad (6.12) \\
& \sum_{\tau=1}^{t} \sum_{\alpha \in I} x_{\alpha \tau} \cdot a_{\alpha p} \geq \sum_{\tau=1}^{t} \sum_{\beta \in O} y_{\beta \tau} \cdot b_{\beta p}, \quad \forall t = 1, 2, \ldots, T; \quad p \in P \quad (6.13) \\
& x_{\alpha t} \in \{0, 1\}, \quad \forall \alpha \in I; \quad t = 1, 2, \ldots, T \quad (6.14) \\
& y_{\beta t} \in \{0, 1\}, \quad \forall \beta \in O; \quad t = 1, 2, \ldots, T \quad (6.15)
\end{align*}

The objective (6.8) minimizes the maximum of the absolute differences of operation times of outbound trucks $\beta$ and inbound trucks $\alpha$. The constraint (6.9) ensures that each inbound truck is processed in exactly one slot, whereas (6.10) enforces that in each slot at most one inbound truck can be assigned. In analogy, these two conditions hold true for outbound trucks by constraints (6.11) and (6.12). Constraints (6.13) ensure that an outbound truck can only be assigned to a slot $t$ whenever all required products are available (delivered by preceding inbound trucks yet not consumed by preceding outbound trucks) to satisfy the demand for product units of each type $p$. So the available stock accumulated by all inbound trucks assigned to slots $\tau = 1, 2, \ldots, t$ has to exceed the total demand for product units of outbound trucks scheduled up to the actual slot $t$ (recall that it will actually be slot $t + m$ or even $t + m + 1$ when realizing the schedule). The constraints (6.14) and (6.15) represent the binary variables for inbound and outbound trucks respectively.
On providing the left shifting property, Boysen et al. [21] established the following proposition:

**Proposition 1**: The TRSP is NP-hard in the strong sense.

The overall TRSP problem is decomposed into two sub-problems in [21], namely inbound and outbound TRSPs, written as IBD-TRSP and OBD-TRSP respectively. It is divided into sub-problems by fixing a particular inbound (outbound) sequence and then finding the optimal outbound (inbound) sequence respectively. A comparison of IBD-TRSP and OBD-TRSP reveals that their mathematical structures are identical. As a consequence, any algorithm for OBD-TRSP can be used to solve IBD-TRSP and vice versa. In fact, IBD-TRSP can be seen as a reverted OBD-TRSP, in the sense that the solution of an instance of IBD-TRSP with an algorithm designed for OBD-TRSP requires the following steps:

1. Revert the given outbound sequence and set it as the modified inbound sequence. Consider the set of inbound trucks $I$ to be scheduled as the modified set of outbound trucks $O$.

2. Solve OBD-TRSP with the modified input data.

3. The reverted optimal outbound sequence constitutes the optimal inbound sequence for the original IBD-TRSP instance.

An exact dynamic programming approach is introduced and a heuristic starting procedure is proposed to solve the identified sub-problems [21]. The algorithmic descriptions are limited to OBD-TRSP, as they are directly transferable to IBD-TRSP. We conjecture that the multi-level JIT production problem (i.e., ORVP) and the cross-docking supply chain logistics (i.e., CD-SCL) problem are counterparts of each other. In this regard, we propose the following proposition to integrate the ORVP and the CD-SCL problem:
Proposition 2: The solution of the ORVP is balanced if and only if the solution of the CDSCL problem is balanced.

Thus, it is clear that to obtain the solution of either problem is the solution of the other one. However, the problem is still open. Though the two problems are dealt by various scientists independently, the simultaneous study of these problem are very nominal so far. Therefore, this is being one of the notable line of research in production and distribution areas.

6.3 On Performance Modeling of Queueing

Queueing theory deals with one of the most unpleasant experiences of life that is waiting. Queueing is quite common in many real life situations, for example, in telephone exchange, in a supermarket, in a bank counter, at a petrol station, at computer systems etc. Therefore it has a very wide area of research dimentiion. In this section, we discuss on the performance analysis of unreliable $M/M/n/n$ queueing system where the first two $M$ denote the arrival rate and service time respectively and the last two $n$ denote the parallel servers and number of customers. This system is a multi server queueing model where customers arrive in a poisson process and service time is distributed exponentially. The system is arranged to serve for 'n' customers. That is, if there are more than 'n' customers, they cannot get the service. We discuss transient analysis of $M/M/n/n$ queueing model with time dependent arrival and service rates wherein we set up the system of transient balance differential - difference equations and solve them for the probability distribution for any time 't' and show the effect of arrival rate, service rate and time over the probability distribution [49].

In several research works in queueing theory, the study of queueing model has been made under the non-failure servers but this is not realistic. In many real life situations, servers remain to be the technical devices such as
RAM in computer, camera in unman aerial vision, sensor in automatic operating machines where devices are subject to break down and such a device is termed as unreliable. In other words, non-reliable server means that the server is typically subject to unpredictable breakdowns. Transient study of multi-server queueing system under the provision of unreliable concept has been found very rare in the literature. Dorda [38] contributed to modeling and simulation of a Markov multi-server queueing system subject to breakdowns and with an ample repair capacity where the system does not form the queue of waiting customers. Ghimire et al. [48] calculated bulk queueing model with the fixed batch size 'b' and has obtained the expressions for mean waiting time in the queue, mean time spent in the system, mean number of customers (work pieces) in the queue. The brief literature review has been recently carried out in [49].

With introducing proper notations and posing some important assumptions needed for our model, we have derived mathematical model of the problem in terms of a system of differential-difference equations. The coefficient matrix of the system has been analysed and the numerical results are interpreted via matlab simulations. We use devices in many realistic situations to provide services to the customers. We have experienced that components of machines are subject to breakdown such as in unman aerial vision, sensor in automatic operating machines. In this condition, service will be affected. This realistic situation under the constraint that whenever server is under breakdown, no service is provided. Moreover, arrival and the service both depend on time and hence we have studied the time-dependent arrival and service rates that makes the model closure to the realistic [49]. Several situations are there in our daily life, where long queues are formed and the customers get the service one after the other, but in this model no long queue will be formed. There are $n$ number of servers to provide the service for $n$ number of customers. If $j$ number of servers is broken down there will be only $(n - j)$ number of customers to get the service. We can still extend the area of study further,
in which, long queue can be formed and customers will wait for the service staying in the queue. If we consider the service for infinite customers that would be more challenging research area. Cyclic queueing in production assembly line is another direction to be studied further.
Chapter 7: Some Applications

Chapter 7 is based on the papers J, K.

Just-in-time sequencing problem and supply chain systems have quite a wide spectrum of applications in real-life situations. Decision making, problem solving, commodity producing, knowledge performing and finished goods distributing are the intensive works of most of the companies today, which require readily available and suitable information. In this chapter, we present the role of information in production and distribution systems, and in addition, we illustrate some of the applications of both the problems that are performed so far in different production industries.

In today's highly competitive and dynamic society, information is considered as one of the important production factors besides capital and human resources [114]. Finding the right information at the right time to support a business decision process is often very difficult. Recent studies show that a growing number of business executives perceive information flow as a serious challenge. Quite recent direction, the Informed Logistics (IL) investigates the approaches and solutions for an improved information supply for all kinds of distributed work environments, such as enterprises with several locations, networks of small and medium enterprises and mobile workforces [117].
Chapter 7

Some Applications

Chapter 7 is based on the papers J, K.

Just-in-time sequencing problem and supply chain systems have quite a wide spectrum of applications in real-life situations. Decision making, problem solving, commodity producing, knowledge performing and finished goods distributing are the intensive works of most of the companies today, which require readily available and suitable information. In this chapter, we present the role of information in production and distribution systems, and in addition, we illustrate some of the applications of both the problems that are performed so far in different production industries.

In today’s highly competitive and dynamic society, information is considered as one of the important production factors besides capital and human resources [114]. Finding the right information at the right time to support a business decision process is often very difficult. Recent studies show that a growing number of business executives perceive information flow as a serious challenge. Quite recent direction, the Informed Logistics (IL) investigates the approaches and solutions for an improved information supply for all kinds of distributed work environments, such as enterprises with several locations, networks of small and medium enterprises and mobile workforces [117].
In this context, modeling and capturing information demand is a key element of IL solutions. Logistics management is the process of planning, implementing and controlling the efficient, cost-effective flow and storage of raw materials, in-process inventory, finished goods and related information from point-of-origin to point-of-consumption targeting to meet requirements of customers. The aim of IL is to deliver the right information, in the right format, at the right place at the right time for the right people in a customer demand driven approach [114], known as a JIT information distribution system. Gaining logistics information excellence in today’s agile industrial scenario has become a boardroom level concern. Logistics performance leaders are building a significant advantage over the competition that leads to substantially better bottom line performance and increases shareholder value relying upon real-time information.

The focus of management society today has shifted from a manual labor society to an information society, where informed decision making has become one of the key productivity factors in business. Most of the organizations largely depend on knowledge and information, which have become part of their assets. Though information is widely made available, for instance by the evolution of the internet, it is harder to retrieve the right information in desired time. So certain structure is needed to deal with information flows among every node of organizations. The collection, distribution and aggregation of information are essential activities directly contributing to the business goals and should therefore be optimized as a part of the business process. A pre-requisite is the full availability, accessibility and the quality of the information to be delivered. However, present management systems have faced their limits: mass production is optimized but customized production is still not efficiently supported. It is one of the considerable challenges as well as a significant opportunity for research.

Information can be created throughout the entire span of logistics in the production process and re-used in a structured manner along the value creation
In this context, modeling and capturing information demand is a key element of IL solutions. Logistics management is the process of planning, implementing and controlling the efficient, cost-effective flow and storage of raw materials, in-process inventory, finished goods and related information from point-of-origin to point-of-consumption targeting to meet requirements of customers. The aim of IL is to deliver the right information, in the right format, at the right place at the right time for the right people in a customer demand driven approach [114], known as a JIT information distribution system. Gaining logistics information excellence in today’s agile industrial scenario has become a boardroom level concern. Logistics performance leaders are building a significant advantage over the competition that leads to substantially better bottom line performance and increases shareholder value relying upon real-time information.

The focus of management society today has shifted from a manual labor society to an information society, where informed decision making has become one of the key productivity factors in business. Most of the organizations largely depend on knowledge and information, which have become part of their assets. Though information is widely made available, for instance by the evolution of the internet, it is harder to retrieve the right information in desired time. So certain structure is needed to deal with information flows among every node of organizations. The collection, distribution and aggregation of information are essential activities directly contributing to the business goals and should therefore be optimized as a part of the business process. A pre-requisite is the full availability, accessibility and the quality of the information to be delivered. However, present management systems have faced their limits: mass production is optimized but customized production is still not efficiently supported. It is one of the considerable challenges as well as a significant opportunity for research.

Information can be created throughout the entire span of logistics in the production process and re-used in a structured manner along the value creation chain. The aim of IL is to optimize the content and format of available data to reduce throughput times and to achieve a high degree of parallel processing. This requires the use of an information model, an overall product tree and a graphic design concept. The deployed system must meet these requirements in optimal way. Upon these facts, IL means to acquire desired data in real-time (i.e., being informed) and to apply them optimally as information and knowledge in production process making intelligent decisions in traffic flows including road, sea and air; that is, data logistics [114, 123].

The developments in information and document management are required from the perspective of the user, and his/her abilities to serve the customer in the best way. The user of information has always been the driver of innovation, dealing with the four facets of information: input (data), storage, processing and output. The particular challenge is related to the efficient uses and movement of the huge amount of data that has been collected. Thus, ultimate goal of IL is to gain real-time data and apply them to achieve maximum control and visibility of fleet and deliveries for better real-time decisions in favor of customers. So efficiently and effectively managed data and information exchange are becoming crucial along the logistics chain.

Various information technology enablers are applied for the proper use of information flow in production supply chain; to name some of them- logistics information system, electronic data interchange, real-time communication capability, bar coding, radio frequency identification (RFID), global positioning systems [114]. A new perspective of IL in managing information is to create unique value through smart and coordinated systems. The operational priorities must be shifted from spending money on information to making money from information.

With emerging globalization of production and distribution, the logistics networks design and transportation planning are becoming critical competitive techniques to optimize the overall supply chain system. So, the logistics net-
works are closely related to transportation problem. The application of information technology to the transportation or logistics problem is known as the intelligent logistics, which incorporates transportation planning, telecommunications, computing, vehicles and electronics manufacturing, and infrastructure construction as well. Green logistics, third-party logistics, city logistics and reverse-logistics are some newly developed areas each of them having rich applications in different situations. Robust optimization is performed to address the data uncertainty as stochastic optimization.

Effectively and efficiently managing and sharing of public sector information has the power to satisfy customer demand, improve society and drive economic growth of nation as a whole. Good governance, professionalism and dynamism in knowledge and information management are the key elements both to capture opportunities and to meet the challenges ahead. Quality information is needed on overall costs and consequences to make informed decisions and operations.

We have given a new approach to handle raw data as systematic information in production logistics which includes four phases of information: Awareness, Building, Sharing among partners and Operating in production systems; to which we call ABSO information model [114]. We refer [99] too for details. The ABSO model converts data into information and then to knowledge in accelerating the informed flow for customers to increase their flexibility and reduce costs, where the level of information hierarchy is: data → information → knowledge → intelligence. Effective supply chain practice and information sharing enhance the current supply chain management environment [114]. The just-in-time logistics improves the supply chain distribution system with the instant delivery of the needed components or products [78]. Effective supply chain practice and information sharing enhance the current supply chain management environment which is an interconnected system as shown in Figure 7.1.
works are closely related to transportation problem. The application of information technology to the transportation or logistics problem is known as the intelligent logistics, which incorporates transportation planning, telecommunications, computing, vehicles and electronics manufacturing, and infrastructure construction as well. Green logistics, third-party logistics, city logistics and reverse-logistics are some newly developed areas each of them having rich applications in different situations. Robust optimization is performed to address the data uncertainty as stochastic optimization. Effectively and efficiently managing and sharing of public sector information has the power to satisfy customer demand, improve society and drive economic growth of nation as a whole. Good governance, professionalism and dynamism in knowledge and information management are the key elements both to capture opportunities and to meet the challenges ahead. Quality information is needed on overall costs and consequences to make informed decisions and operations.

We have given a new approach to handle raw data as systematic information in production logistics which includes four phases of information: Awareness, Building, Sharing among partners and Operating in production systems; to which we call ABSO information model [114]. We refer [99] too for details. The ABSO model converts data into information and then to knowledge in accelerating the informed flow for customers to increase their flexibility and reduce costs, where the level of information hierarchy is: data → information → knowledge → intelligence. Effective supply chain practice and information sharing enhance the current supply chain management environment [114]. The just-in-time logistics improves the supply chain distribution system with the instant delivery of the needed components or products [78]. Effective supply chain practice and information sharing enhance the current supply chain management environment which is an interconnected system as shown in Figure 7.1.

Next we discuss on some applications of JIT sequencing problem and supply chain in different industrial areas. As explained earlier, the main application of JITPS has been originally performed in Toyota company (Japan) by producing products efficiently and keeping costs low. Some of the most successful companies in the world have used JIT philosophy to improve their manufacturing processes and better meet customer demand. Here we briefly present the applications JITPS and supply chain in some other industries of different countries.

The construction performance are improved in United Kingdom through
greater standardisation of components, more off-site pre-fabrication and the use of lean construction techniques. The Commonwealth of Australia advised that construction firms use the newest technologies and management processes including JIT techniques, ensuring the Australian construction industry remains competitive [82]. In the Chinese ready mixed concrete (RMC) industry, it is observed that all RMC suppliers were practising JIT procurement of cement, long-term relationships with the contractors, group technology and other JIT principles. In the industry of batching plants in Singapore, it is observed that the JIT purchasing system is feasible to procure the raw materials, which can significantly reduce the amount of buffer stock on site [82]. It is noteworthy that the initiatives and support from the government on new management techniques are significant factors in the improvement of the construction industries.

There have been conducted two case studies in Ethiopian industries as the application of JITPS - the first is in automotive manufacturing company, and the second one is in bottle and glass factory [47]. These studies show that the implementation of JITPS could significantly reduce the waste and trained human resource could understand the JIT principle to improve the productivity. Similarly it is found that JITPS is implemented in a small company in Taiwan that produces different kinds of automobile lamps such as rear combination lamps and front turn signal lamps.

Dell manufacturing company has also cashed the JIT principles to uplift its manufacturing process to a success, which is able to provide exceptionally short lead times to their customers, by forcing their suppliers to carry inventory instead of carrying it themselves and then demanding (and receiving) short lead times on components so that products can be simply assembled by Dell quickly and then directly shipped to the customer. Dell carried 20 to 25 days of inventory in a sprawling network of warehouses before one and half decade. Now a days, it has no warehouses. And though it assembles nearly 80,000 computers every 24 hours, it carries no more than 2 hours of
greater standardisation of components, more off-site pre-fabrication and the use of lean construction techniques. The Commonwealth of Australia advised that construction firms use the newest technologies and management processes including JIT techniques, ensuring the Australian construction industry remains competitive [82]. In the Chinese ready mixed concrete (RMC) industry, it is observed that all RMC suppliers were practising JIT procurement of cement, long-term relationships with the contractors, group technology and other JIT principles. In the industry of batching plants in Singapore, it is observed that the JIT purchasing system is feasible to procure the raw materials, which can significantly reduce the amount of buffer stock on site [82]. It is noteworthy that the initiatives and support from the government on new management techniques are significant factors in the improvement of the construction industries.

There have been conducted two case studies in Ethiopian industries as the application of JITPS - the first is in automotive manufacturing company, and the second one is in bottle and glass factory [47]. These studies show that the implementation of JITPS could significantly reduce the waste and trained human resource could understand the JIT principle to improve the productivity. Similarly it is found that JITPS is implemented in a small company in Taiwan that produces different kinds of automobile lamps such as rear combination lamps and front turn signal lamps. Dell manufacturing company has also cashed the JIT principles to uplift its manufacturing process to a success, which is able to provide exceptionally short lead times to their customers, by forcing their suppliers to carry inventory instead of carrying it themselves and then demanding (and receiving) short lead times on components so that products can be simply assembled by Dell quickly and then directly shipped to the customer. Dell carried 20 to 25 days of inventory in a sprawling network of warehouses before one and a half decade. Now a days, it has no warehouses. And though it assembles nearly 80,000 computers every 24 hours, it carries no more than 2 hours of inventory in its factories and a maximum of just 72 hours across its entire operation. Michael Dell, the CEO of the company, is fond of saying that in the high-tech business, you either grow or die.

In the same way, Harley Davidson’s use of JITPS in the late World War II era is mostly characterized by its transformation from an inefficient manufacturer with extra inventory to a nimble manufacturer able to meet demand and provide short lead times. This implementation of JITPS increased productivity and decreased inventory levels by 75 percent; that is, the inefficiencies in the production processes were quickly identified and solved.

The other effective use of JITPS and supply chain management is performed by Wal-Mart. JIT ordering enables Wal-Mart company to decrease the costs associated with inefficient inventory decisions and handling. The replacement of bar-code technology by RFID enables Wal-Mart to improve the efficiency of its global supply chain management through greater supply chain visibility and more accurate ordering decisions. To make its distribution process more efficient, Wal-Mart also made use of cross-docking logistics techniques which reduced the handling and storage of finished goods, virtually eliminating the role of the distribution centers and stores.

All information related to sales and inventories was passed on through an advanced satellite communication system. It is noted that as a effort to implement new technologies to reduce costs and increase the efficiency, Wal-Mart asked its top 100 suppliers in July 2003 to be RFID compliant by January, 2005. Thus, the company used both the cross-docking and the information technologies to enhance its business and to satisfy customers with quality products at demanded time. There can be found other many manufacturing companies that implemented the JITPS and cross-docking supply chain logistics around the globe.
Chapter 8
Conclusion, Discussion and Further Works

In this dissertation, we have carried out an extensive account of just-in-time production system (JITPS) and its advanced theory of sequencing approaches in mixed-model production systems with the goal of keeping the rate of usage of parts as constant as possible. We have characterized the just-in-time sequencing problems (JITSP) via discrete apportionment approach and further pointed out the synchronization of production and distribution problems. The mixed-model JITSP has been developed around 1970s and the supply chain system has been developed around 1990s. Both the problems have real-life applications in manufacturing industries, which has been briefly discussed in Chapter 7.

The mixed-model JITSP has been widely and extensively studied with various mathematical models and aspects in the literature [20, 31, 33, 112, 128]. In spite of these studies, the problem has still remained challenging to be studied further since it has interesting mathematical base model of theoretical value and wide real-world applications. A number of open problems related with mixed-model JITSP has been listed in this dissertation for further research work.
Chapter 8

Conclusion, Discussion and Further Works

In this dissertation, we have carried out an extensive account of just-in-time production system (JITPS) and its advanced theory of sequencing approaches in mixed-model production systems with the goal of keeping the rate of usage of parts as constant as possible. We have characterized the just-in-time sequencing problems (JITSP) via discrete apportionment approach and further pointed out the synchronization of production and distribution problems. The mixed-model JITSP has been developed around 1970s and the supply chain system has been developed around 1990s. Both the problems have real-life applications in manufacturing industries, which has been briefly discussed in Chapter 7.

The mixed-model JITSP has been widely and extensively studied with various mathematical models and aspects in the literature [20, 31, 33, 112, 128]. In spite of these studies, the problem has still remained challenging to be studied further since it has interesting mathematical base model of theoretical value and wide real-world applications. A number of open problems related with mixed-model JITSP has been listed in this dissertation for further research work.
The discrete apportionment problem (DAP) has a long history and different facades to apportion the integer seats to any state or party according to the population of the state or according to the number of votes. Among various methods of apportionment [6, 112], we have applied the divisor methods to mathematically charaterize the JITSP and developed two divisor methods proposing an upper bound for the objective of total PRVP. In this research work, the noble interlink of JITSP and DAP has been carried out via local and global deviation approaches and pointed out some open problems in this direction too.

The single-level case of mixed-model JITSP, the product rate variation problem (PRVP), has been solved in pseudo-polynomial time. It is noteworthy that the total PRVP with a general objective and the bottleneck PRVP only with the absolute-deviation objective have been solved in the literature. The total PRVP has been mathematically reduced to the assignment problem defining ideal position and assignment cost, and hence solved in $O(D^3)$ time. The bottleneck PRVP with absolute-deviation objective has been solved by reducing into an order-preserving perfect matching for feasible solution and then using a bisection search algorithm for optimality in $O(D \log D)$ time. In addition, it has been solved by reducing into the bottleneck assignment problem and solved in $O(D^3)$ time in the literature [60].

The synthetic study to minimize the inequality measures of the PRVP and DAP simultaneously has been carried out in this work [111]. However, there exist no exact solution to obtain equality between actual and ideal productions, though some methods are devised to minimize the inequality between actual and ideal quantities in production and DAP. We have established equitably efficient solution for the two problems and further extended the approach in [34] discussing in complexity issues. We have proposed local deviation approach of DAP to minimize local inequality of PRVP and global deviation approach of DAP to minimize global inequality of product to product rate variation problem. It remains open that whether balancing the local
The discrete apportionment problem (DAP) has a long history and different facades to apportion the integer seats to any state or party according to the population of the state or according to the number of votes. Among various methods of apportionment [6, 112], we have applied the divisor methods to mathematically characterize the JITSP and developed two divisor methods proposing an upper bound for the objective of total PRVP. In this research work, the noble interlink of JITSP and DAP has been carried out via local and global deviation approaches and pointed out some open problems in this direction too.

The single-level case of mixed-model JITSP, the product rate variation problem (PRVP), has been solved in pseudo-polynomial time. It is noteworthy that the total PRVP with a general objective and the bottleneck PRVP only with the absolute-deviation objective have been solved in the literature. The total PRVP has been mathematically reduced to the assignment problem defining ideal position and assignment cost, and hence solved in \( O(D^3) \) time.

The bottleneck PRVP with absolute-deviation objective has been solved by reducing into an order-preserving perfect matching for feasible solution and then using a bisection search algorithm for optimality in \( O(D \log D) \) time. In addition, it has been solved by reducing into the bottleneck assignment problem and solved in \( O(D^3) \) time in the literature [60].

The synthetic study to minimize the inequality measures of the PRVP and DAP simultaneously has been carried out in this work [111]. However, there exist no exact solution to obtain equality between actual and ideal productions, though some methods are devised to minimize the inequality between actual and ideal quantities in production and DAP. We have established equitably efficient solution for the two problems and further extended the approach in [34] discussing in complexity issues. We have proposed local deviation approach of DAP to minimize local inequality of PRVP and global deviation approach of DAP to minimize global inequality of product to product rate variation problem. It remains open that whether balancing the local measures of inequality indicate balancing the global measures of the inequality.

In spite of various noble efforts to solve the PRVP with polynomial time complexity in the input size of the demands, the exact complexity of the problem still remains open. The problem has been proved to be Co-NP but remains open whether it is Co-NP complete or polynomially solvable. Thoughtful analysis of the work-in-progress till the date shows that the solution to the problem with polynomial time complexity seems unlikely to exist.

The multi-level case of mixed-model JITSP, the output rate variation problem (ORVP), has been proved to be NP-hard. However, there exist some sequencing heuristics including dynamic programming for near optimal solution and pegging concept to reduce the problem into the weighted case of PRVP, as explained in Section 4.2. The existence of cyclic optimal sequences considerably reduces the computational time in case of PRVP. However, the conjecture whether the cyclic sequences are optimal for the ORVP is still open.

For the bottleneck deviation not exceeding the bound \( B \), a JIT sequence exists in single-level if and only if \((i, j)\) takes position \( k \) in the sequence such that \( E(i, j) \leq k \leq L(i, j) \) [73]. The bottleneck analysis is important to reduce the early-tardy penalties. It is established that an optimal sequence for \( F_{max}^a \) always exists when the deviation for each product is less than or equal to one and greater than or equal to \( 1 - r_{max} \) [103], where \( r_{max} \) is the largest ideal production rate \( r_i \) among all models. That is, \( B = 1 \) is an upper bottleneck and \( 1 - r_{max} \) is the lower bottleneck for the problem such that \( 1 - r_{max} \leq \) optimal value \( \leq 1 \). The lower bottleneck is tight which can not always be obtained. However, if it is attained for an instance then it is always optimal. In the other end, the upper bottleneck is not tight and hence there can be found some modifications and improvements on this bottleneck in the literature.
The lower and upper bottlenecks defined above are slightly modified for the weighted min-max objective function \( \max_{i,k} w_i |x_{ik} - kr_i| \) in [26]. The lower bottleneck on the optimal value for this problem is \( \min_i w_i(1 - r_i) \) and the upper bottleneck is \( \max_i w_i \). The formulas for the earliest and latest starting times are also modified with the weight factor to construct the bipartite graph. We have mentioned that \( B = 1 - \frac{1}{D} \) is also an upper bottleneck given in [25] which clearly implies that the upper bound \( B = 1 \) is not tight.

Since the bottleneck \( 1 - \frac{1}{D} \) is also not tight, we are further left with the problem to find the tighter upper bottleneck for the problem. However, we have proposed one tighter bottleneck \( 1 - \frac{1}{\sqrt{D}} \) in this dissertation with two conjectures. The new bound significantly reduces the deviation and the bisection search is performed within the smaller interval \( [1 - r_{\max}, 1 - \frac{1}{\sqrt{D}}] \) which lowers the time complexity. The algebraic approach of balanced words introduced for absolute bottleneck PRVP partially address the problem. The 1-balanced words cannot be obtained for most rates, however the set of all 3-balanced words consists of optimal sequence to the problem. The minimality of set of 3-balanced words is unknown and enumeration of this set is expensive for optimality. It is still open whether the set of all 2-balanced words is sufficient for an optimal sequence for problem \( F_{\text{max}}^a \).

We have put forward the issue that the ORVP can be dealt jointly with the supply chain logistics problem which can balance the multi-facade chain of overall production planning problem. Supply chain logistics (SCL) is a bit broader concept in production planning. Certain barriers exist to deter the research in this field. With the expansion of the supply chain to a global status, the average researchers are left dry with little or no access to the corporate data, and with little resource to look into the status and devise solutions. Due to this reason, most research works have been limited to individual aspects of SCL with limited data, and mostly has been empirical research. This shows that the research community of operations research needs to find a way around this problem.

150
The geographic distances among the different parts of a supply chain system poses another problem. A normal researcher can work at one location; but some financial and diplomatic barriers often prevent the researcher from accessing the other parts of a global supply chain. This indicates that the global supply chain of modern competitive time should be addressed by cross-country collaboration. We further intend to handle the multi-level supply chain logistics problem via input-output matrix approach. In addition, both the problems PRVP and ORVP can be analysed in case of the stochastic situation which is also our due course.

The theoretical researches in applied mathematics in Nepal appear to be limited in library and classroom activities. And the industries are also not using proper mathematical modeling to optimize their operations. Therefore, the prime challenge of today’s research work is to link up the academic research with industrial operations. The just-in-time sequencing of products in single-level and multi-level can be applied in the production industries of Nepal, for instance in the industries of sugar, noodles, distillary and confectionary. Our further aim is to continue the research simultaneously in production and distribution areas with the goal to collaborate the academic and industrial areas in Nepal. This requires group effort.

Coming together is a beginning,
Keeping together is progress,
Working together is success.

→ Henry Ford
References


References


[38] Dorda M (2010) Modelling and simulation of unreliable M/M/n/n queueing system, Electronic technical magazine of technology, engineering and logistic in transport, **5(4)**:37-45.


[57] Jost V (2003), Deux problemes d’approximation diophantibe: Le patage propoirional en nombres entries et les pavages equilibres de Z. DEA ROCO, Laboratoire Leibniz- IMAG.


Kubiak W (1992) Completion time variance minimization on single machine is difficult, working paper 92-6, Memorial University of Newfoundland.


Kubiak W (1992) Completion time variance minimization on single machine is difficult, working paper 92-6, Memorial University of Newfoundland.


[92] Morse, Marton, Newmann JV and Eisenhart LP (1948) Report to the President of the National Academy of Sciences, USA, 28 May 1948.


[109] Tambe PY (2006) Balancing mixed-model assembly line to reduce workload in a multi-level production system, Dr. BAM University, India.


Index

3-partition problem, 37
Absolute deviation, 27, 54, 93
Actual cumulative production, 20
Actual production, 54
Algorithms, 39
Apportionment, 16
Apportionment paradoxes, 17
Assembly line, 2, 42
Assignment cost, 79
Assignment problem, 72
Balanced word, 66, 150
Bicriteria sequencing, 87
Bipartite graph, 58
Bisection search, 62
Bottleneck, 3, 21, 82
Bottleneck product rate variation problem, 21, 117
Cardinality, 54
Change-over cost, 2
Circuit satisfiability, 36
Co-NP, 63, 72
Co-NP-complete, 63, 72
Combinatorial problem, 36
Competing, 15
Complexity, 28, 149
Computational time, 82
Concatenation, 79
Congressional apportionment, 95
Constraint, 21
Convex bipartite graph, 70, 82
Cross-docking operation, 123
Cross-docking operation, 126
Cumulative production, 54
Cyclic sequences, 79
Decision science, 15
Deviation, 3
Discrete apportionment problem, 15, 26
Divisor function, 96
Divisor methods, 1, 96, 148
Downstream distribution, 31
Dynamic programming, 45, 52, 134, 149
Earliest due date, 28, 42
Earliest sequencing time, 69
Earlyness, 42
Efficient frontier, 108
Electoral systems, 16
Equitably efficient, 4, 106, 148
Even instances, 80
Extended goal chasing method, 50
Fair division, 15
Fair queueing algorithms, 119
Fair sequence, 65
Fairness measure, 96
Feasibility, 77
Feasible schedules, 77
Feasible sequence, 72, 85
Final assembly sequencing, 44
Final level, 49
Finishing time, 119
Finite word, 66
Floor, 92
Folding, 80
Global deviation, 104
Glover’s EDD rule, 70
Greatest common divisor, 81
Hall’s theorem, 71
Hamilton method, 96
Hard real-time, 116
Heuristic algorithms, 40
Heuristics, 39, 48, 56
House monotone, 97
House of representatives, 15
House size, 92
Hungarian method, 77
Huntington method, 96
Hybrid algorithms, 108
Ideal position, 103
Ideal production, 3, 20, 54
Impossibility theorem, 93
Inbound logistics, 31, 125
Inequality measures, 99
Information technology, 141
Informed logistics, 139
Integer programming, 3, 21
Intelligent logistics, 142
Inventories, 2, 9
JIT purchasing, 121
JIT sequencing, 14
Just-in-time, 1, 8, 10
Just-in-time delivery, 121
Just-in-time production system, 2, 8
Kanban, 11
Latest sequencing time, 69
Level curve, 57, 69
Lexicographic, 44
Liu-Layland periodic scheduling, 116
Loading operations, 127
Local deviation, 106
Logistics, 123
Lower bottleneck, 60, 85
Matrix representation, 54
Maximum deviation, 3
Maximum matching, 70
Mean-based divisors, 102
Measure of inequality, 98
Min-max objectives, 30, 35
Min-sum objectives, 30, 35
Minimum weight perfect matching, 84
Mixed-model assembly line, 2
Mixed-model sequencing, 9, 13
Model, 2, 3
Modified EDD rule, 43
Monge inequality, 79
Monotone methods, 96
Monotonicity, 21
Nonlinear integer programming, 35
NP-hard, 3, 36, 51, 118, 149
NP-hard problems, 36
Oneness property, 83, 85, 86
Optimal schedules, 35
Optimal solution, 62, 78, 79
Optimality, 29, 44, 61, 77
Optimization algorithm, 82
Ordering, 13, 28
Outbound logistics, 31, 125
Output rate variation problem, 2, 30, 53
Overproduction, 35
Pairwise comparison, 98
Pairwise equity, 16
Pairwise transfer, 97
Parametric methods, 100
Pareto algorithm, 87
Pegging, 51
Penalty function, 72
Perfect matching, 56
Periodic word, 66
Permutation, 13
Polynomial time, 82, 149
Probability, 39
Processing time, 37, 43
Product, 2
Product level, 49
Product rate variation problem, 2, 20, 51, 65
Proportional representation, 108
Proportionally, 16
Pseudo polynomial heuristic, 28
Pseudo-polynomial, 3, 37, 51, 63
Pull systems, 4, 10
Queueing, 135
Rank-index, 96
Real life problems, 15
Relative difference, 103
Response time variability problem, 114
Rounding, 92
Scheduling around the shortest job, 36
Sequencing, 2, 34
Sequencing cost, 73
Sequencing problem, 13
Shortages, 2, 9
Shuffle, 80
Single-machine scheduling, 28
Square deviation, 27, 36, 55
Starting time, 119
Steiner tree problem, 36
Sub-assembly level, 49
Supply chain, 124
supply chain management, 113
Swapping, 80
Switch-over cost, 3
Symmetric output rate variation problem, 56
Symmetry, 80

Tardiness, 42
Tie-and-transfer, 108
Time complexity, 40, 82
Time window, 70
Total deviation, 3
Total product rate variation problem, 22
Toyota production system, 8
Tractability, 27
Truck sequencing problem, 129

Underproduction, 35
Unfolding, 80

Upper bottleneck, 60, 85
Upstream production, 31
Weighted product rate variation problem, 52